**TRAINING OBJECTIVE**

The main objective of the Industrial Training is learn new technology so that we are able work in the organization and competes with the world. Understand and learning new technology in **a Computer Science** is very important. A good training could be really beneficial for us in the future.

Also getting a industrial training from a recognized and experience company help us to understand real life situations in industrial organizations and their related environments and accelerating the learning process of how student’s knowledge could be used in a realistic way.

It provides the exposure to practice and apply the acquired knowledge “hands - on” in the working environment. Industrial training also provides a systematic introduction to the ways of industry and developing talent and attitudes, so that one can understand how Human Resource Development works. Moreover, students can gain hands-on experience that is related to the students majoring so that the student can relate to and widen the skills that have been learnt while being in university. Industrial training also exposes the students to the real career world. Furthermore, students implement what they have learned and learn more throughout this training. Besides, students can also gain experience to select the optimal solution in handling a situation. During industrial training students can learn the accepted safety practices in the industry.

**BRIEF OF ORGANIZATION**

Tech Hive is a Multi Domain Organization, covering almost all major trends of modern day technology. From the day of our establishment, we have been constantly widening our horizons and spreading out our feathers to meet the ever increasing demands of our clients. We deal basically internationally and nationally according to demand of our clients. We feel immense honor to introduce ourselves as one of the leading- Embedded , Industrial Automation , Smart Software based Application and Product Development Company. Tech Hive is a web development company and consulting firm specializing in business & management consulting, IT solutions, web site design & development and Software development. Our teams of consultants are able to provide complete business solutions to clients locally as well as internationally.

We also specialize in ERP systems and custom software development designed according to your business specifications. Whatever your needs, whether you are a small business or a large corporation, our Web Design company will provide you with solutions and designs to suit your budget and requirements. Our highly qualified team of consultants have over five years’ experience in providing quality consulting & guidance as well as offering our expertise in software, design and development. Our developers are qualified to work on C/C++, .net, JAVA , Oracle , web designing , PHP, Drupal, Joomla

**INTRODUCTION**



**PhotoStudio** is GUI based desktop application developed in php. Photo Studio application can be very helpfull for everyone to keep the in photos in an organized way and saved. After clicking the photos, user can access any time anywhere by logine to our website. Other than that, by the method picture are saved in cloud so chance of loss of personals photos are less.

**OBJECTIVE OF THE PROJECT**

To help and extend the manual business of photo-studio we decided to provide a photos of customers online.

* Time saving.
* Always accessiable.
* Easy to find photos.

**MODULES AND THEIR DESCRIPTION**

1. Index
2. User
   1. Login
   2. Show Photos
   3. Select Albums
3. Admin
   1. Add user
   2. Add Admin
   3. Add Album
   4. Add Photos
4. **Main**

In this page customer can see the portfolio of photo studio. Also customer can see the about us and contact us page.

1. **User**

User Module is used by the user to login. User is required start this application. Every user must have a user name and password for login.

1. **Admin**

This Module use to add the user and add photos.

**Add user-**: This Option can be helpful to create the user to provide password that is required for successful login.

**HARDWARE AND SOFTWARE REQUIREMENTS**

**Hardware:** Hardware is the physical part of the computer system like mouse, keyboard, monitor etc.

**HARDWARE REQUIREMENTS:**

1. Any PC processor

2. 512 MB Ram

3. Keyboard mouse

4. Internet Connection

**Software:** Software is a set of applications which is used to run the operating system.

The software requirements to successfully run all the programs are:

**Brief description of these component**

* **HTML:-**  
  **HTML**, which stands for **Hyper Text Markup Language**, is the predominant markup language for web pages HTML is the basic building-blocks of webpages.
* **JavaScript:-**  
  **JavaScript**, also known as ECMA Script is a prototype-based object oriented scripting language that is dynamic, weakly typed and has first class functions. It is also considered a functional programming language like scheme and O Caml because it has closures and supports higher order functions.
* **CSS:-**  
  **Cascading Style Sheets** (**CSS**) is a style sheet language used to describe the presentation semantics (the look and formatting) of a document written in a markup language.
* **BOOTSTRAP(CSS):-**

Bootstrap is a powerful front-end framework for faster and easier web development. It includes HTML and CSS based design templates for common user interface components like Typography, Forms, Buttons, Tables, Navigations, Dropdowns, Alerts, Modals, Tabs, Accordion, Carousel and many other as well as optional JavaScript extensions.

Bootstrap also gives us an ability to create responsive layout with much less efforts.

* **PHP:-**  
  **PHP** is a general-purpose scripting language originally designed for web development to produce dynamic web pages. This language is best to begin with for web development.
* **MySql**  
  MySql is a relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases. Lightweight and effective database.
* **XAMPP server 2.0(for winows platform)**XAMPP server is free to use for all the developers, XAMPP server runs on only windows based platform, if we used different OS then we have to used different version of XAMPP i.e(lamp for linux, mamp for mac os), xamp is common PHP based development tools for all perating system.
* **Filezilla**

It is a FTP(File transfer Protocol) used to upload the file/project to host site. It is ensure to upload a project to the host server through filezilla and it is very fast to upload the file.

**NECCESSARY TO SOFTWARE AND HARDWARE**

**Software**

Software, commonly known as programs, consists of all the electronic instructions that tell the hardware how to perform a task. These instructions come from a software developer in the form that will be accepted by the operating system that they are based on. For example, a program that is designed for the Windows operating system will only work for that operating system. Compatibility of software will vary as the design of the software and the operating system differ. Software that is designed for Windows XP may experience compatibility issue when running under Windows 7.

Software can also be described as a collection of routines, rules and symbolic languages that direct the functioning of the hardware. Software is capable of performing specific tasks, as opposed to hardware which only perform mechanical tasks that they are mechanically designed for. Practical computer systems divide software systems into three major classes:

* **System software**: Helps run computer hardware and computer system. Computer software includes operating systems, device drivers, diagnostic tools and more.
* **Programming software**: Software that assists a programmer in writing computer programs.
* **Application software**: Allows users to accomplish one or more tasks.

**Software Requirements**

**Development Platform**

Server : Apache, Xampp

Front End : Core PHP

Data Base : MYSQL

**Working Platform**

Operating System : Microsoft Windows 2000 and more

Data Base : MYSQL

**Hardware**

Hardware refers to the physical elements of a computer. Also referred to as the machinery or the equipment of the computer. Examples of hardware in a computer are the keyboard, the monitor, the mouse and the processing unit However, most of a computer's hardware cannot be seen; in other words, it is not an external element of the computer, but rather an internal one, surrounded by the computer's casing. A computer's hardware is comprised of many different parts, but perhaps the most important of these is the motherboard. The motherboard is made up of even more parts that power and control the computer.

In contrast to software, hardware is a physical entity, while software is a non-physical entity. Hardware and software are interconnected, without software; the hardware of a computer would have no function. However, without the creation of hardware to perform tasks directed by software via the central processing unit (box), software would be useless. Hardware, in the computer world, refers to the physical components that make up a computer system. There are many different kinds of hardware that can be installed inside, and connected to the outside, of a computer. Some of common hardware that find inside a computer are Motherboard, Central Processing Unit (CPU), Random Access Memory (RAM), Power Supply, Video Card, Hard Drive, Optical Drive (i.e. BD/DVD/CD drive), Sound Card, Network Interface Card (NIC), Analog Modem, FireWire/USB Expansion Card, etc. Some of common hardware that might find connected to the outside of a computer is Keyboard, Mouse, Printer, Scanner, Speakers, Monitor, etc. The following hardware is referred to as network hardware and various pieces are often part of a home or business network such as Router, Network Switch, Access Point, Repeater, Bridge, Print Server, Firewall, etc

Software Requirements: -

* **Operating System :** Windows
* **Web-Technology: PHP**
* **Front-End: HTML, CSS (Bootstrap), JAVASCRIPT**
* **Back-End:** MySQL
* **Web Server:** Apache SERVER.

**ABOUT PHP**

PHP is an HTML-embedded scripting language. Much of its syntax is borrowed from C, Java and Perl with a couple of unique PHP-specific features thrown in. The goal of the language is to allow web developers to write dynamically generated pages quickly.

**PHP** is a server-side scripting language designed for web development but also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in 1994,[4] the PHP reference implementation is now produced by The PHP Group.[5] PHP originally stood for *Personal Home Page*,[4] but it now stands for the recursive backronym *PHP: Hypertext Preprocessor*.[6]

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 command-line interface (CLI) and can be used to implement standalone graphical applications.[7]

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.[8]

## What is a PHP File?

* PHP files can contain text, HTML, CSS, JavaScript, and PHP code
* PHP code are executed on the server, and the result is returned to the browser as plain HTML
* PHP files have extension ".php"

### OBJECT ORIENTED

Object-Oriented PHP helps your code stay flexible by allowing it to be only defined once but used in many places.

**ANALYSIS**

**FEASIBILITY STUDY:-**

A feasibility study is a preliminary study which investigates the information needs of perspective users and determines the resource requirements, determining the cost effectiveness of various alternatives in the designs of the information system, benefits and feasibility of proposed project.

The goal of the feasibility study is to evaluate alternative systems to propose the most feasible and desirable systems for development.

**The feasibility of our proposed system can be evaluated as:-**

1. **TECHNICAL FEASIBILITY.**
2. **SCHEDULE FEASIBILITY.**
3. **OPERATIONAL FEASIBILITY.**
4. **ECONOMIC FEASIBILITY.**

**TECHNICAL FEASIBILITY:-**

Technical feasibility can be demonstrated if reliable hardware and software capable of meeting needs of proposed system can be developed or acquired by the business in required time. Our project is technically feasible because the required hardware and software needed for our project are available.

**SCHEDULE FEASILBILTY: -**

The scheduled feasibility of the system evaluates whether the system finishes its task within the provided time of the development or not. Our project is feasible according to schedule because our project is being completed within the specified time period of our semester.

**OPERATIONAL FEASIBILITY: -**

Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. In the manual system, it is very difficult to maintain huge amount of pricing information of products. The development of the new system was started because of the requirements put forward by the management of the concerned department. So it is sure that the system development is operationally feasible.

**ECONOMIC FEASIBILITY:-**

Here the development cost is evaluated by weighing it against the ultimate benefits derived from the new system,. The benefit accrued from the new system is more than the cost involved in its development as everything is related to money.

The proposed system is economically feasible because the cost involved in purchasing that hardware and the software are within approachable. The operating –environment costs are marginal .The less time involved also helped in its economical feasibility.

Feasibility study is a test of system proposal according to its workability, impact on the organization, ability to meet user needs, and effective use of resources.

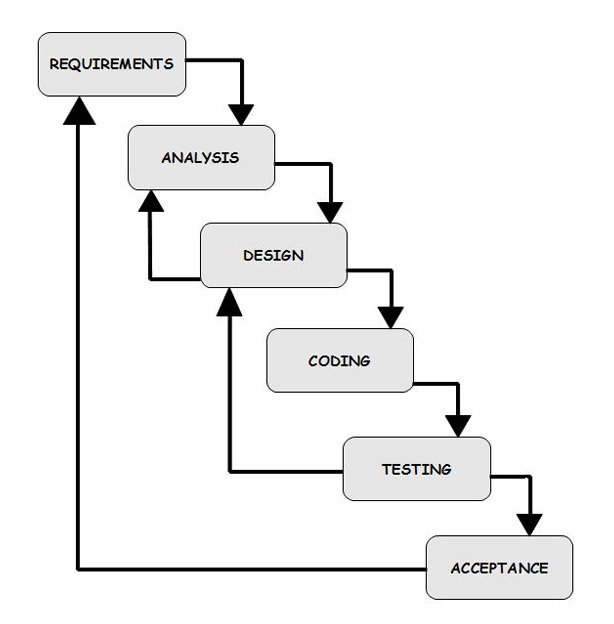
The objective of a feasibility study is not to solve the problem but to acquire a sense of its scope. During the study, the problem definition is crystallized and aspects of the problem to be included in the system are determined. Consequently, cost and benefits are estimated with the greater accuracy at this stage.

The result of the feasibility study is a formal proposal. This is simply a report — a formal document detailing the nature and scope of the proposed solution. The proposal summarizes what is known and what is going to be done.

**SOFTWARE MODEL**

**Waterfall Model:-**

The waterfall model is a sequential design process, often used in software development processes, in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of Conception, Initiation, Analysis, Design, Coding, Testing, Implementation, and Maintenance .

****

The Waterfall Model (System Development Life Cycle)

In waterfall model, the following phases are followed in order:

1. Requirements specification
2. Feasibility Study
3. Design & Coding
4. Integration & Testing
5. Implementation
6. Maintenance

**Requirement Analysis & Definition:** All possible requirements of the system to be developed are captured in this phase. Requirements are set of functionalities and constraints that the end-user (who will be using the system) expects from the system. The requirements are gathered from the end-user by consultation, these requirements are analyzed for their validity and the possibility of incorporating the requirements in the system to be development is also studied. Finally, a Requirement Specification document is created which serves the purpose of guideline for the next phase of the model.

**System & Software Design:** Before a starting for actual coding, it is highly important to understand what we are going to create and what it should look like? The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture. The system design specifications serve as input for the next phase of the model.

**Implementation & Unit Testing:** On receiving system design documents, the work is divided in modules/units and actual coding is started. The system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality; this is referred to as Unit Testing. Unit testing mainly verifies if the modules/units meet their specifications.

**Integration & System Testing:** As specified above, the system is first divided in units which are developed and tested for their functionalities. These units are integrated into a complete system during Integration phase and tested to check if all modules/units coordinate between each other and the system as a whole behaves as per the specifications. After successfully testing the software, it is delivered to the customer.

**Operations & Maintenance:** This phase of "The Waterfall Model" is virtually never ending phase (Very long). Generally, problems with the system developed (which are not found during the development life cycle) come up after its practical use starts, so the issues related to the system are solved after deployment of the system. Not all the problems come in picture directly but they arise time to time and needs to be solved; hence this process is referred as Maintenance.

**MYSQL**

Mysql is a relational database management system, which organizes data in the form of tables. Mysql is one of many database servers based on RDBMS model, which manages a seer of data that attends three specific things-data structures, data integrity and data manipulation.

With Mysql cooperative server technology we can realize the benefits of open, relational systems for all the applications. Mysql makes efficient use of all systems resources, on all hardware architecture; to deliver unmatched performance, price performance and scalability. Any DBMS to be called as RDBMS has to satisfy Dr.E.F.Codd’s rules.

**FEATURES OF MYSQL**

**PORTABLE**

The Mysql RDBMS is available on wide range of platforms ranging from PCs to super computers and as a multi user loadable module for Novel NetWare, if you develop application on system you can run the same application on other systems without any modifications.

**COMPATIBLE**

Mysql commands can be used for communicating with IBM DB2 mainframe RDBMS that is different from Mysql, which is Mysql compatible with DB2. Mysql RDBMS is a high performance fault tolerant DBMS, which is specially designed for online transaction processing and for handling large database applications.

**MULTITHREADED SERVER ARCHITECTURE**

Mysql adaptable multithreaded server architecture delivers scalable high performance for very large number of users on all hardware architecture including symmetric multiprocessors (sumps) and loosely coupled multiprocessors. Performance is achieved by eliminating CPU, I/O, memory and operating system bottlenecks and by optimizing the mysql DBMS server code to eliminate all internal bottlenecks.

Mysql has become the most popular RDBMS in the market because of its ease of use

* Client/server architecture.
* Data independence.
* Ensuring data integrity and data security.
* Managing data concurrency.
* Parallel processing support for speed up data entry and online transaction processing used for applications.
* DB procedures, functions and packages.

**Dr.E.F.CODD’S RULES**

These rules are used for valuating a product to be called as relational database management systems. Out of 12 rules, a RDBMS product should satisfy at least 8 rules + rule called rule 0 that must be satisfied.

**RULE 0: FOUNDATION RULE**

For any system to be advertised as, or claimed to be relational DBMS should manage database with in itself, without using an external language

**RULE 1: INFORMATION RULE**

All information in relational database is represented at logical level in only one way as values in tables.

**RULE 2: GUARANTEED ACCESS**

Each and every data in a relational database is guaranteed to be logically accessibility by using to a combination of table name, primary key value and column name.

**RULE 3: SYSTEMATIC TREATMENT OF NULL VALUES**

Null values are supported for representing missing information and inapplicable information. They must be handled in systematic way, independent of data types.

**RULE 4: DYNAMIC ONLINE CATALOG BASED RELATION MODEL**

The database description is represented at the logical level in the same way as ordinary data so that authorized users can apply the same relational language to its interrogation as they do to the regular data.

**RULE 5: COMPREHENSVE DATA SUB LANGUAGE**

A relational system may support several languages and various models of terminal use. However there must be one language whose statement can express all of the following DataDefinitions,ViewDefinitions, DataManipulations,Integrity,Constraints,Authorization .

**RULE 6: VIEW UPDATING**

Any view that is theoretical can be updatable if changes can be made to the tables that effect the desired changes in the view.

**RULE 7: HIGH LEVEL UPDATE, INSERT AND DELETE**

The capability of handling a base relational or derived relational as a single operand applies not only retrieval of data also to its insertion, updating, and deletion.

**RULE 8: PHYSICAL DATA INDEPENDENCE**

Application program and terminal activities remain logically unimpaired whenever any changes are made in either storage representation or access method.

**RULE 9: LOGICAL DATA INDEPENDENCE**

Application programs and terminal activities remain logically unimpaired whenever any changes are made in either storage representation or access methods.

**RULE 10: INTEGRITY INDEPENDENCE**

Integrity constraints specific to particular database must be definable in the relational data stored in the catalog, not in application program.

**RULE 11: DISTRIBUTED INDEPENDENCE**

Whether or not a system supports database distribution, it must have a data sub-language that can support distributed databases without changing the application program.

**RULE 12: NON SUB-VERSION**

If a relational system has low level language, that low language cannot use to subversion or by pass the integrity rules and constraints expressed in the higher level relational language.

**MYSQL SUPPORTS THE FOLLOWING CODD’S RULES**

Rule 1: Information Rule (Representation of information)-YES.

Rule 2: Guaranteed Access-YES.

Rule 3: Systematic treatment of Null values-YES.

Rule 4: Dynamic on-line catalog-based Relational Model-YES.

Rule 5: Comprehensive data sub language-YES.

Rule 6: View Updating-PARTIAL.

Rule 7: High-level Update, Insert and Delete-YES.

Rule 8: Physical data Independence-PARTIAL.

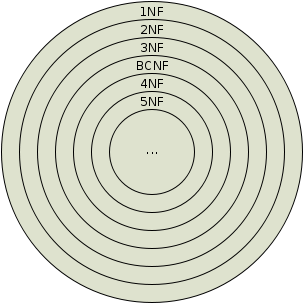
Rule 9: Logical data Independence-PARTIAL.

Rule 10: Integrity Independence-PARTIAL.

Rule 11: Distributed Independence-YES.

Rule 12: Non-subversion-YES

**NORMALIZATION**

****

A Database is a collection of interrelated data stored with a minimum of redundancy to serve many applications. The database design is used to group data into a number of tables and minimizes the artificiality embedded in using separate files. The tables are organized to:

* Reduced duplication of data.
* Simplify functions like adding, deleting, modifying data etc..,
* Retrieving data
* Clarity and ease of use
* More information at low cost

Normalization is built around the concept of normal forms. A relation is said to be in a particular normal form if it satisfies a certain specified set of constraints on the kind of functional dependencies that could be associated with the relation. The normal forms are used to ensure that various types of anomalies and inconsistencies are not introduced into the database.

**FIRST NORMAL FORM**

A relation R is in first normal form if and only if all underlying domains contained atomic values only.

**SECOND NORMAL FORM**

A relation R is said to be in second normal form if and only if it is in first normal form and every non-key attribute is fully dependent on the primary key.

**THIRD NORMAL FORM**

A relation R is transitively depends on the primary key. said to be in third normal form if and only if it is in second normal form and every non key attribute is non transitively depend on the primary key.

**In BCNF:**

Though the relation is in 3NF, 3NF does not deal satisfactorily with the case of a relation with overlapping candidate keysi.e. composite candidate keys with at least one attribute in common.

BCNF is based on the concept of a determinant.A determinant is any attribute (simple or composite) on which some other attribute is fully functionally dependent.

A relation is in BCNF is, and only if, every determinant is a candidate key.

When a relation has more than one candidate key, anomalies may result even**:** A relation is in Boyce-Codd Normal Form (BCNF) if every determinant is a candidate key. (See the links in the box at right for definitions of determinant and candidate key.)  
Whereas BCNF insists that for this dependency to remain in a relation, A must be a candidate key.

**Fourth Normal Form (4th NF)**  
An entity is in Fourth Normal Form (4NF) when it meets the requirement of being in Third Normal Form (3NF) and additionallyhas no multiple sets of multi-valued dependencies. In other words, 4NF states that no entity can have more than a single one-to-many relationship within an entity if the one-to-many attributes are independent of each other.

Fourth Normal Form applies to situations involving many-to-many relationships.In relational databases, many-to-many relationships are expressed through cross-reference-tables.  
  
 A table is in fourth normal form (4NF) if and only if it is in BCNF and contains no more than one multi-valued dependency.

**Fifth Normal Form (5th NF)**  
  
A relation that has a join dependency cannot be decomposed by a projection into other relations without spurious results

A relation is in 5NF when its information content cannot be reconstructed from several smaller relations i.e. from relations having fewer attributes than the original relation

Fifth normal form, also known as join-projection normal form (JPNF), states that no non-trivial join dependencies exist. 5NF states that any fact should be able to be reconstructed without any anomalous results in any case, regardless of the number of tables being joined. A 5NF table should have only candidate keys and it's primary key should consist of only a single column.

**SYSTEM ANALYSIS**

**EXISTING SYSTEM & DISADVANTAGES**

In the existing system, all the work done manually.

* It is very time consuming
* It is Very Complex
* It requires a lot of man force.

**PROPOSED SYSTEM & ITS ADVANTAGES**

In the proposed system the all the work done automatically by our software, we just have to enter the detail of work.

* Less time Consuming
* Result generate by the system is very accurate.
* Less Complex
* Requires less man force.

**SYSTEM DESIGN**

Systems design is the process or art of defining the architecture components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. There is someoverlap and synergy with the disciplines of systems analysis, systems architecture and systems engineering.

**UML {UNIFIED MODELING LANGUAGE}**

The Unified Modeling Language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules..A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

**USER MODEL VIEW**

This view represents the system from the users perspective.The analysis representation describes a usage scenario from the end-users perspective.

**STRUCTURAL MODEL VIEW**

In this model the data and functionality are arrived from inside the system.This model view models the static structures.

**BEHAVIORAL MODEL VIEW**

It represents the dynamic of behavoriral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

**IMPLEMENTATION MODEL VIEW**

In this the structural and behavioural as parts of the system are represented as they are to be built.

**ENVIRONMENTAL MODEL VIEW**

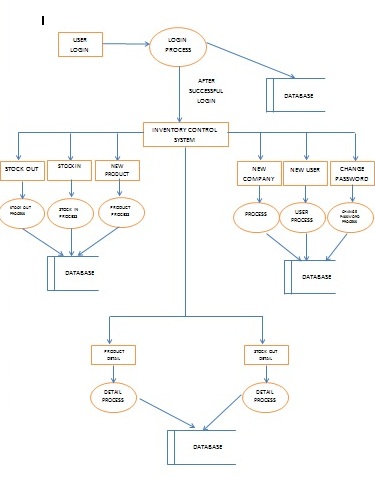
In this the structural and behavioural aspects of the environment in which the system is to be implemented are represented. UML is specifically constructed through two different domains they are:

UML Analysis modelling, this focuses on the user model and structural model views of the system.UML design modelling, which focuses on the behaviouralmodelling, implementation modelling and environmental model views.

Use case Diagrams represent the functionality of the system from a user’s point of view. Use cases are used during requirements elicitation and analysis to represent the functionality of the system. Use cases focus on the behavior of the system from external point of view.

Actors are external entities that interact with the system. Examples of actors include users like administrator, bank customer …etc., or another system like central database.

**DATA FLOW DIAGRAMS**

****

**SYSTEM TESTING**

****

Testing is a process, which reveals errors in the program. It is the major quality measure employed during software development. During software development. During testing, the program is executed with a set of test cases and the output of the program for the test cases is evaluated to determine if the program is performing as it is expected to perform.

**TESTING IN STRATEGIES**

In order to make sure that the system does not have errors, the different levels of testing strategies that are applied at differing phases of software development are:

**UNIT TESTING**

Unit Testing is done on individual modules as they are completed and become executable. It is confined only to the designer's requirements. Each module can be tested using the following two Strategies:

**BLACK BOX TESTING**

In this strategy some test cases are generated as input conditions that fully execute all functional requirements for the program. This testing has been uses to find errors in the following categories:

* Incorrect or missing functions
* Interface errors
* Errors in data structure or external database access
* Performance errors
* Initialization and termination errors.

In this testing only the output is checked for correctness. The logical flow of the data is not checked.

**WHITE BOX TESTING**

In this the test cases are generated on the logic of each module by drawing flow graphs of that module and logical decisions are tested on all the cases. It has been uses to generate the test cases in the following cases:

* Guarantee that all independent paths have been Executed.
* Execute all logical decisions on their true and false Sides.
* Execute all loops at their boundaries and within their operational bounds
* Execute internal data structures to ensure their validity.

**INTEGRATING TESTING**

Integration testing ensures that software and subsystems work together a whole. It tests the interface of all the modules to make sure that the modules behave properly when integrated together.

**ACCEPTANCE TESTING**

It is a pre-delivery testing in which entire system is tested at client's site on real world data to find errors.

**TEST APPROACH**

Testing can be done in two ways:

* Bottom up approach
* Top down approach

**BOTTOM UP APPROACH**

Testing can be performed starting from smallest and lowest level modules and proceeding one at a time. For each module in bottom up testing a short program executes the module and provides the needed data so that the module is asked to perform the way it will when embedded with in the larger system. When bottom level modules are tested attention turns to those on the next level that use the lower level ones they are tested individually and then linked with the previously examined lower level modules.

**TOP DOWN APPROACH**

This type of testing starts from upper level modules. Since the detailed activities usually performed in the lower level routines are not provided stubs are written. A stub is a module shell called by upper level module and that when reached properly will return a message to the calling module indicating that proper interaction occurred. No attempt is made to verify the correctness of the lower level module.

**VALIDATION**

The system has been tested and implemented successfully and thus ensured that all the requirements as listed in the software requirements specification are completely fulfilled. In case of erroneous input corresponding error messages are displayed

**IMPLEMENTATION**

**OVERVIEW OF IMPLEMENTATION**

Implementation is the process of having systems personnel check out and put new equipment into use, train users, install the new application and construct any files of data needed to use it. This phase is less creative than system design. Depending on the size of the organization that will be involved in using the application and the risk involved in its use, systems developers may choose to test the operation in only one area of the firm with only one or two persons. Sometimes; they will run both old and new system in parallel way to compare the results. In still other situations, system developers stop using the old system one day and start using the new one the next. The implementation phase is less creative than system design. It is primarily with user training, site preparation and file conversion. When the candidate system is linked to terminals or remote sites, the telecommunication network and test of the network along with system are included under the implementation. During final testing, user acceptance is tested, followed by user training. Depending on the nature of the system, extensive user training may be required. Conversion usually takes place at about the same time the user is being trained.

Evaluation of the system is performed to identify its strengths and weaknesses.The actual evaluation can occur along any of the following dimensions:-

1. **Operational Evaluation:**Assessment of the manner in which the system functions, including case of use, response time, overall reliability and level of utilization.
2. **Organizational Impact:**Identification and measurement of benefits to the organization in such areas as financial concerns, operational efficiency and competitive impact.
3. **User ManagerAssessment:**Evaluation of the attitudes of seniors and user manager within the organization, as well as end-users.

**IMPLEMENTATION TOOLS:**

* Training of personnel
* Conversion procedures
* Post-implementation review or evaluation

**Training of personnel:**

The high quality training is an essential step in systems implementation. Hence to provide the training to personnel they have been provided user manuals. They are asked to read it carefully and same thing try practically on computer. If they don’t understand anything, can ask without any hesitation. They are asked to enter the data, which is more frequently entered, and print the reports are frequently printed. Users are told about those situations, which they must understand and should able to handle it.

**Conversion Method:**

The direct conversion method is applied. This method converts from old to the new system abruptly. The old system is used till a planned conversion day. The organization relies fully on the new system.

**Evaluation Or Post Implementation Review:**

After the system is implemented and conversion is complete, a review is conducted to determine whether system is meeting expectations and where improvements are needed. A post implementation review measures the systems performance against pre-determined requirements. It determines how well the system continues to meet performance specifications. It also provides information to determine whether major re-design or modification is required. In evaluation system is checked against the pre-determined requirements. All the requirements have been fully attained.

**REVIEW PLAN:**

For review a team is planned. The team shall prepare a formal plan around the objectives of the review. An overall plan covers the following areas:-

* **Administrator Plan:** Review area objectives, operating costs, actual operating performance and benefits.
* **Personnel Requirements Plan:** Review performance objectives and training performance to data.
* **Hardware Plan**: Review performance specifications
* **Documentation Review Plan**: Review the system development efforts

**Administrative Plan**

**User Objective**: This is an extremely crucial area since it may be possible that over a period of time the system does not meet the initial objectives of the user or objectives get changed as a result of changes in the overall objectives of the organization. The results of the evaluation are documented for future reference.

Operating costs and benefits: Under the administration plan, current budget designed manipulate costs and savings of the system will closely reviewed.

**Personnel Requirement Plan**

Personnel old performance objectives will be compared with current performance levels. Training performance through testing, conducting interviews and other data gathering techniques.

**Hardware Plan**

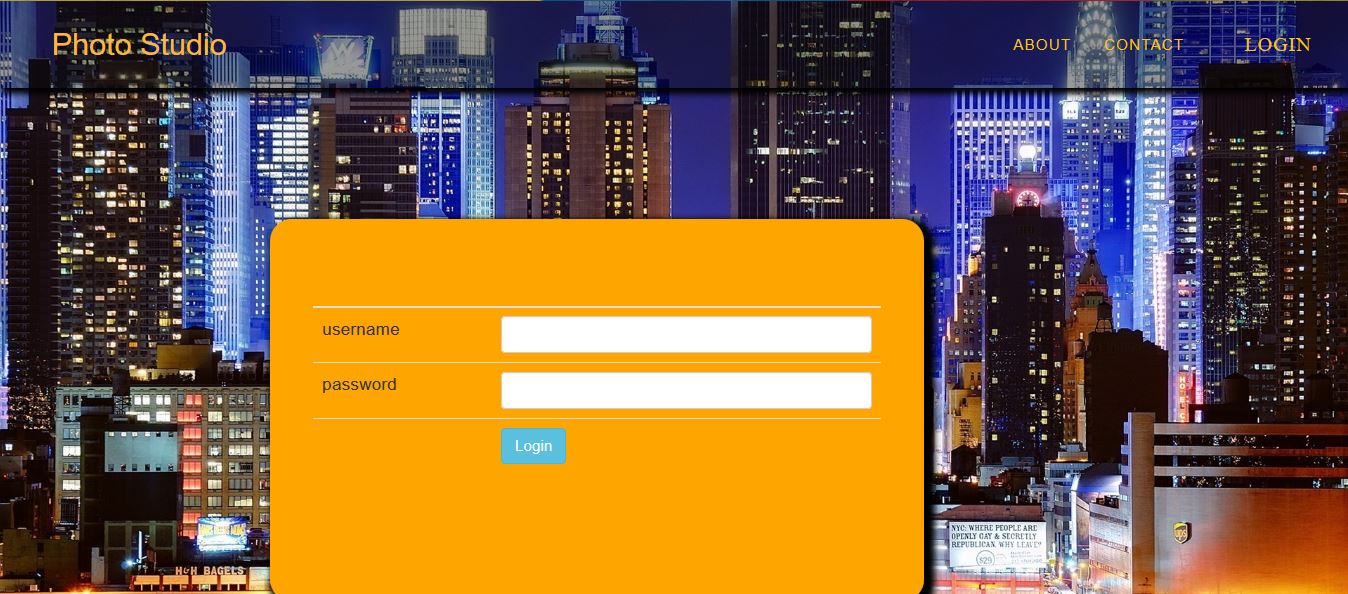
The hardware of the new system will also review. But as we are using existing hardware which fulfill the requirement of the project. Hence the cost of hardware is not increased.

**SCREEN SHOTS**

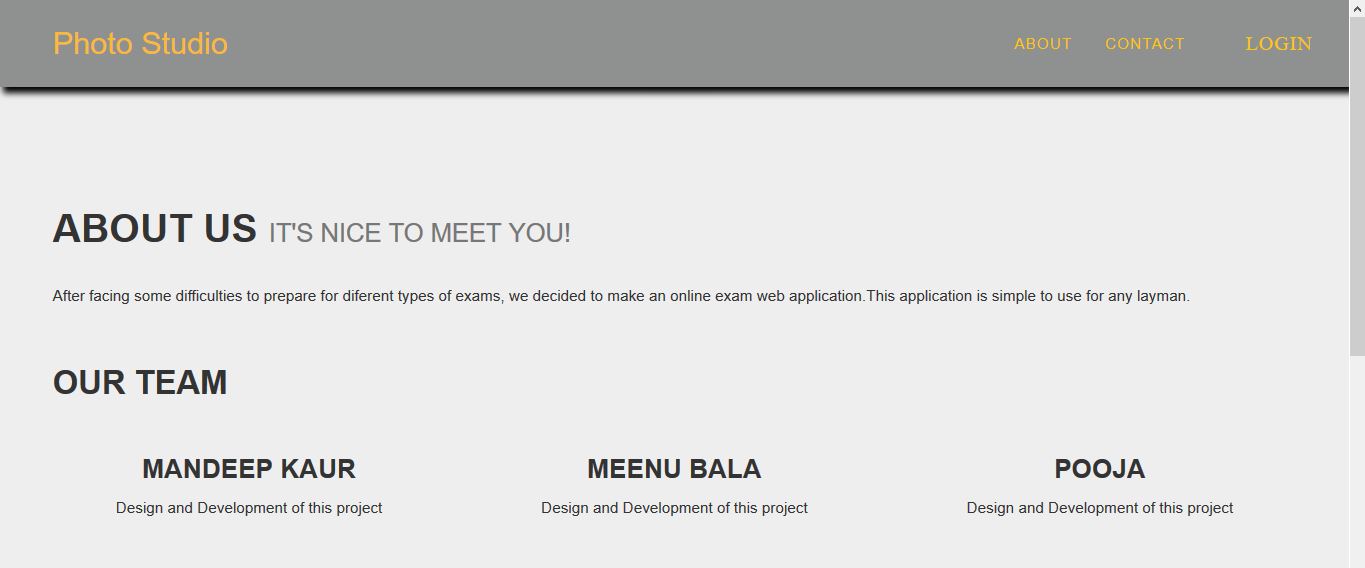
**Home**

****

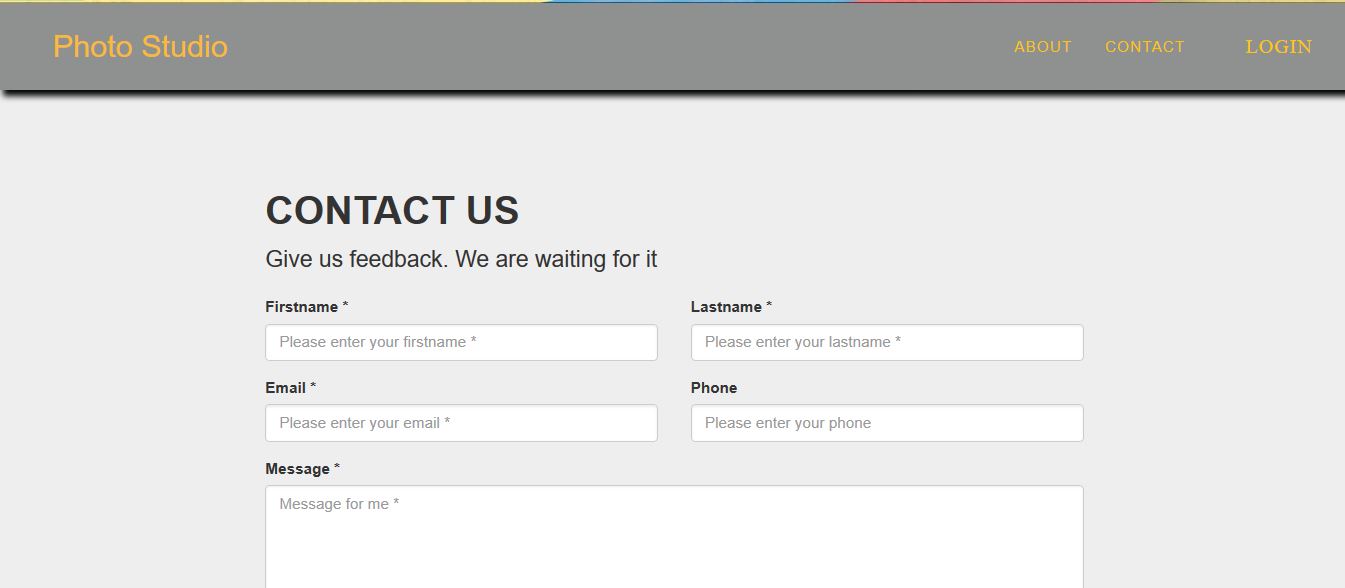
**User LOGIN PAGE**

****

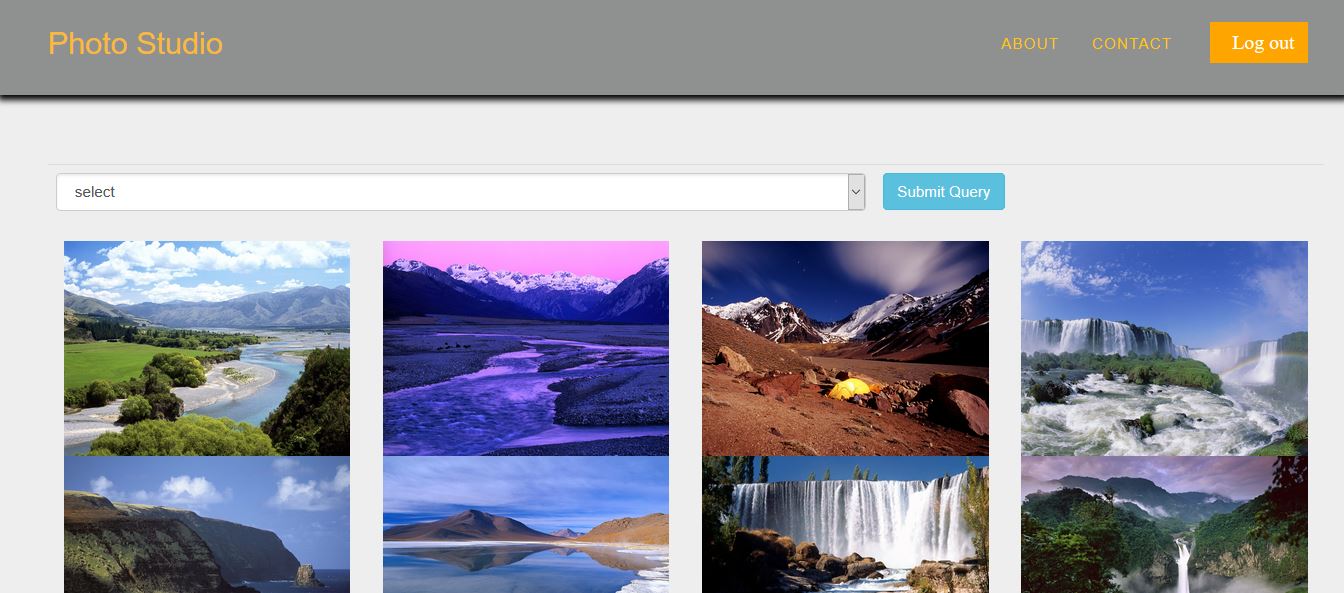
**ABOUT PAGE**



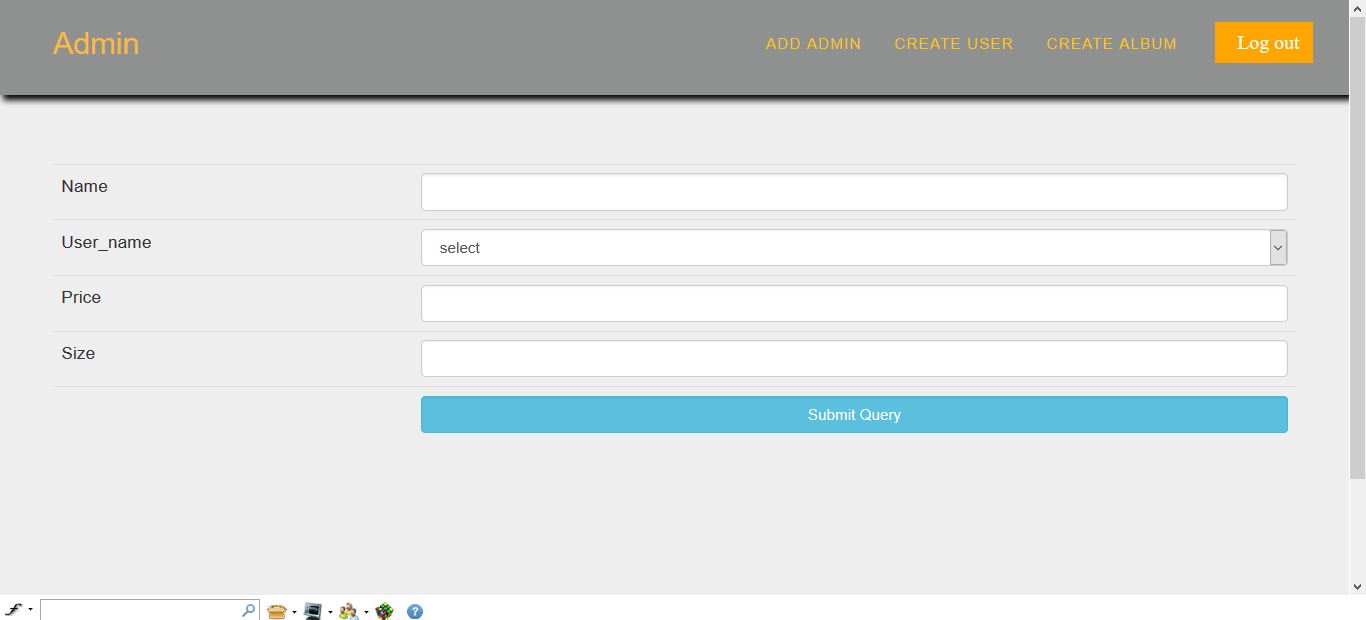
**CONTACT PAGE**



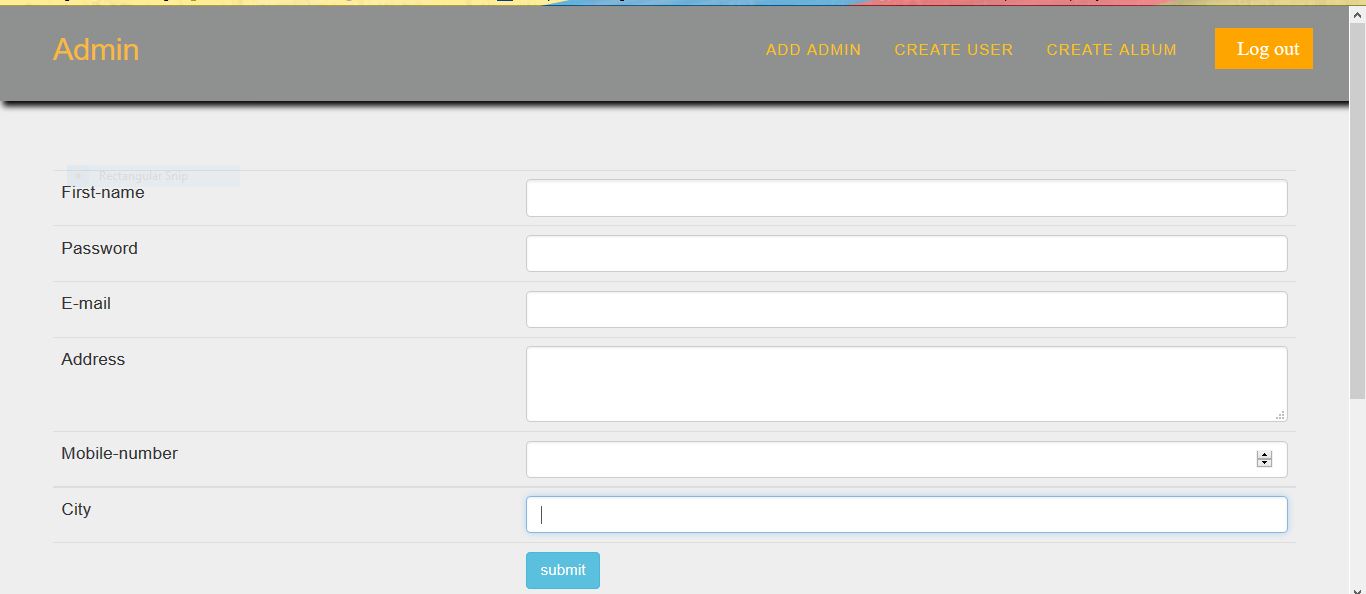
**SHOW PHOTOS**

****

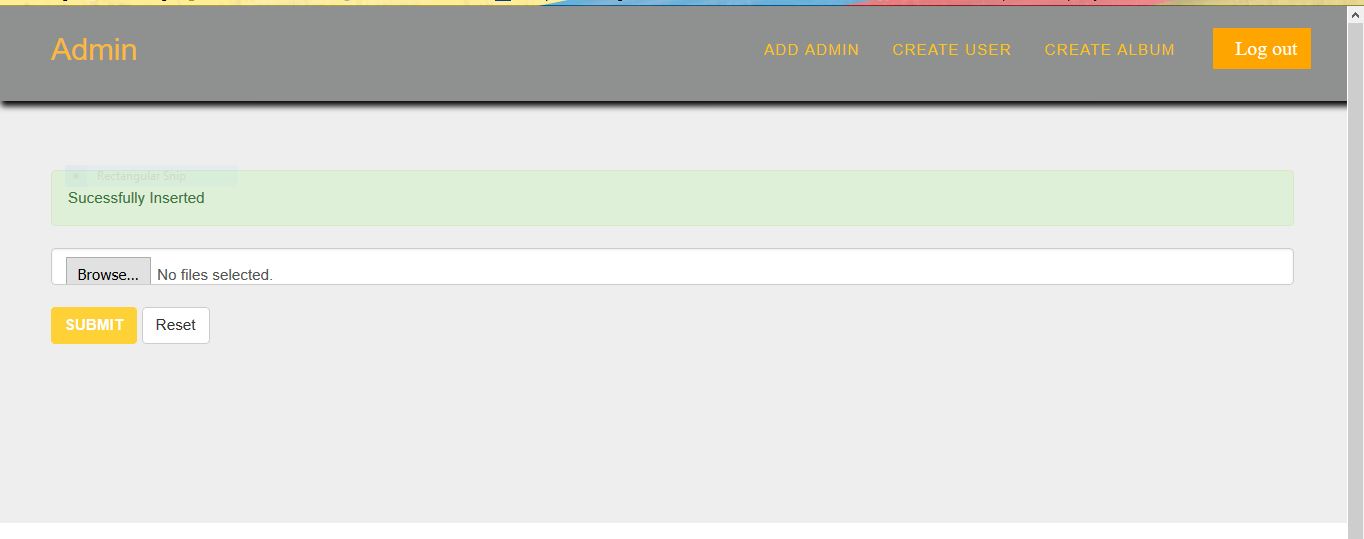
**CREATE ALBUM**

****

**CREATE USER**

****

**ADD PHOTOS**

****

**MAINTENANCE**

**Introduction to Software Maintenance:** Software maintenance denotes any changes to a software product after it has been delivered to the customer. Maintenance is inevitable for almost any kind of product. It is practically impossible to make the software completely error free because the input domain of most software products is very large and it is not practical to test the software exhaustively with respect to each value that the input data may assume. Maintenance is also needed to enhance the features of the software to add more functionality to it and to port to new platforms etc.

**Types of Software Maintenance**

The requirement of software maintenance arises on account of the three main reasons:-

1. **Corrective:**Corrective Maintenance of a software product becomes necessary to rectify the bugs observed while the system is in use.
2. **Adaptive:**A software product might need maintenance when the customers need the product to run on new platforms, on new operating systems, or when they need the product to be interfaced with new hardware or software.
3. **Perfective:**Asoftware product needs maintenance to support the new features that users want it to support, to change different functionalities of the system according to customer demands, or to enhance the performance of the system.

**FUTURE ENHANCEMENT**

Even though the project full fills the requirements of the present application there is always scope for further work. According to the emerging changes and new versions, further work can be done to improve the application since project is designed in flexible software.

Our current application is work only for one warehouse at time, we can make it more advance that only single application is work for all the warehouses of the organization all over the country or world.