**CHAPTER 1**

**INTRODUCTION**

Zoological gardens provide an opportunity to open a whole new world of curiosity and interest, and sensitize visitors regarding the value and need for conservation of wildlife. Zoos were initially started for the entertainment of people. Gradually over the years, they have come to play an important role in conservation. The goal of zoos is the conservation of animals in the wild.

The world in the 21st century is growing up in the technology in every field such as education, medicine, transport etc. The use of technology makes the world so faster and easier than the early world and it releases the world from manual usage in every field.

In the early days, the manual usage causes many mistakes by the user and administrative. Using manual properties in the fields was not comfortable for the consumers because it was slower than technical usages, caused wastages of the consumers’ time and contained many formalities in usage.

“Zoo Management System” is this project of using technology in zoological parks for administration, maintenances, and entrance ticketing fields. Now also most of the zoological parks are using manual programs in their administration, maintenances and ticketing fields. This project was prepared for using computerized programs for the above fields. In that zoological park they used manual programs for administration, maintenance and ticketing field for long years. According to the basic needs of programs for this zoological park, this project was prepared to make the works easier.

**CHAPTER 2**

**PROBLEM STATEMENT**

The main problem and motivation are basically lack of information about various animals. This project is based on the various information related to the animal and the workers, which are present in the zoo. Such as when we visit any zoo then we are not aware that which animal is placed at which place. For knowing this information, we look for a graph present in the zoo, but with this information system we can easily search the zoo by filling the name of the animal, such as Lion, Tiger, etc. then by running this query one can easily find the block in which the animal is present. The system sends an auto-generated order to suppliers when the stock of food and medicine reach the deadline

**CHAPTER 3**

### TECHNICAL FEASIBLITY

### Requirement Analysis:

During the Requirement analysis we analyze our whole proposed system that whether it is possible to develop the proposed system or not. That means we check that it is technically. Economically feasible to develop proposed system or not.

#### Feasibility Study:

A feasibility study is defined as an evaluation or analysis of the potential impact of a proposed project or program. The feasibility study is based on extensive research on both the current practices and the proposed project/program and its impact on the software industry. The feasibility study will contain extensive data related to financial software impact and will include advantages and disadvantages of both the current situation and the proposed plan.

#### Technical Feasibility:

Our main purpose is to make dynamic web pages using PHP&MYSQL. All designing within our system depends on UML concept. We gather information about UML from book and internet. So, we gathered all technical requirements and make our system technically feasible.

#### Economic Feasibility:

Actually, the financial investment is very less for building this proposed system .The only financial investment that will be incurred of the hardware and software like PHP required for development and testing. Hence it is financially liable to build such a proposed system.

* **Operational Feasibility:**

Operational feasibility is nothing but the relation of the speed of the system with respect to time. We have to check whether our system is operational feasible or not. Initially they use to have Peer-to-Peer communication. But now-a-days, Client-Server Architecture is used commonly. As it consumes less time and maintenance cost is reduced. In Client Server, there is no need of keeping a record in a database and increasing the cost of it. As a separate log files are maintained and record for individual clients is maintained. Thus, the operational feasibility is mentioned.

**HARDWARE & SOFTWARE REQUIREMENTS**:

* **Hardware:**
* **Processor:** Core2Duo
* RAM: - 128MB and above
* Hard Disk: - 20GB
* **Software:**
* OS: - Windows.
* Browser: - Google chrome, Mozilla, any JavaScript enabled browser.
* Server: - xampp server.

* **Proposed Project Input:**

Since it is regarding text mining so whatever the input will be there, it will be in the form of text (Query). The query will be regarding the Tickets & Animal Data after that we have produce Related data like animal breed ,animal name, Animal image, and Tickets Type ,Total visitors Basically, the project included datasets also that will Facility the Admin & Staff to Get Data.

* **Proposed Project Output:**

When user enter the system, user find out the Animal Data like Breed, Name,

Cage No, Animal image also Ticket type, Data about visitor, ticket price

User also register new user delete user, edit user details.

#### CHAPTER 4

#### SYSTEM DESIGN IMPLEMENTAION

## SOFTWARE DEVLOPEMENT LIFE CYCLE

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process. The following figure is a graphical representation of the various stages of a typical SDLC.

A typical Software Development Life Cycle consists of the following stages –



**Fig:-**software Development life cycle.

### STAGE 1: PLANNING AND REQUIREMENT ANALYSIS

Requirement analysis is the most important and fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational and technical areas.

Planning for the quality assurance requirements and identification of the risks associated with the project is also done in the planning stage. The outcome of the technical feasibility study is to define the various technical approaches that can be followed to implement the project successfully with minimum risks.

### STAGE 2: DEFINING REQUIREMENTS

Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts. This is done through an **SRS (Software Requirement Specification)** document which consists of all the product requirements to be designed and developed during the project life cycle.

### STAGE 3: DESIGNING THE PRODUCT ARCHITECTURE

SRS is the reference for product architects to come out with the best architecture for the product to be developed. Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in a DDS - Design Document Specification. This DDS is reviewed by all the important stakeholders and based on various parameters as risk assessment, product robustness, design modularity, budget and time constraints, the best design approach is selected for the product. A design approach clearly defines all the architectural modules of the product along with its communication and data flow representation with the external and third party modules (if any). The internal design of all the modules of the proposed architecture should be clearly defined with the minutest of the details in DDS.

### STAGE 4: BUILDING OR DEVELOPING THE PRODUCT

In this stage of SDLC the actual development starts and the product is built. The programming code is generated as per DDS during this stage. If the design is performed in a detailed and organized manner, code generation can be accomplished without much hassle. Developers must follow the coding guidelines defined by their organization and programming tools like compilers, interpreters, debuggers, etc. are used to generate the code. Different high level programming languages such as C, C++, Pascal, Java and PHP are used for coding. The programming language is chosen with respect to the type of software being developed.

### STAGE 5: TESTING THE PRODUCT

This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However, this stage refers to the testing only stage of the product where product defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

### STAGE 6: DEPLOYMENT IN THE MARKET AND MAINTENANCE

Once the product is tested and ready to be deployed it is released formally in the appropriate market. Sometimes product deployment happens in stages as per the business strategy of that organization. The product may first be released in a limited segment and tested in the real business environment (UAT- User acceptance testing).

Then based on the feedback, the product may be released as it is or with suggested enhancements in the targeting market segment. After the product is released in the market, its maintenance is done for the existing customer base.

.

* **SOFTWARE DEVLOPEMENT MODEL**

**WATERFALL MODEL:**

The Waterfall Model was the first Process Model to be introduced. It is also referred to as a **software development life cycle model**. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

The Waterfall model is the earliest SDLC approach that was used for software development.

The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap.

## WATERFALL MODEL

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

The following illustration is a representation of the different phases of the Waterfall



**Fig**: waterfall model

The sequential phases in Waterfall model are −

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

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

## 

## DATA FLOW DIAGRAM

**DFD** is the abbreviation for **Data Flow Diagram**. The flow of data of a system or a process is represented by DFD. It also gives insight into the inputs and outputs of each entity and the process itself. DFD does not have control flow and no loops or decision rules are present. Specific operations depending on the type of data can be explained by a flowchart. Data Flow Diagram can be represented in several ways. The DFD belongs to structured-analysis modeling tools. Data Flow diagrams are very popular because they help us to visualize the major steps and data involved in software-system processes.

**DFD LEVEL 1**

Request For login Request

Admin

User

Response Response

**Fig:** 1 Zoo Management System

**DFD LEVEL 1.1**

Admin

Display

Login

Ticket’s data

**Fig 2**:-1.1:Level DFD for Zoo Management System.

**DFD LEVEL1.2**

Ticketing

Display

**Fig 3:**-1.2: Level DFD for Ticketing.

**DFD LEVEL 2**

Animals Details

Display

**Fig 4:**-2: Level DFD for Ticketing.

* **ADMIN CASE DIAGRAM**

**Fig-5**: Admin case diagram.

* **USER CASE DIAGRAM**

**Fig-5**: User case diagram.

**CHAPTER 5**

**DATABASE DESIGN**

**MYSQL INTRODUCTION**

MySQL pronounced either "My S-Q-L" or "My Sequel," is an open source relational database management system. It is based on the structure query language ([SQL](http://techterms.com/definition/sql)), which is used for adding, removing, and modifying information in the database. Standard SQL commands, such as ADD, DROP, INSERT, and UPDATE can be used with MySQL.

MySQL can be used for a variety of applications, but is most commonly found on Web servers. A website that uses MySQL may include Web pages that access information from a database. These pages are often referred to as "dynamic," meaning the content of each page is generated from a database as the page loads. Websites that use dynamic Web pages are often referred to as database-driven websites.

Many database-driven websites that use MySQL also use a Web scripting language like [PHP](http://techterms.com/definition/php) to access information from the database. MySQL commands can be incorporated into the PHP code, allowing part or all of a Web page to be generated from database information. Because both MySQL and PHP are both open source (meaning they are free to download and use), the PHP/MySQL combination has become a popular choice for database-driven websites.

* **Permission Table:-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SR NO | FIELD NAME | DATA TYPE | SIZE | CONSTAINT |
| 1 | Id | Int | 11 | NOT NULL |
| 2 | Permission | Varchar | 255 | NOT NULL |
| 3 | Createuser | Varchar | 255 | NOT NULL |
| 4 | Deleteuser | Varchar | 255 | NOT NULL |
| 5 | Createbid | Varchar | 255 | NOT NULL |
| 6 | Updatebid | Varchar | 255 | NOT NULL |

* **Admin Table:-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SR NO | FIELD NAME | DATA TYPE | SIZE | CONSTAINT |
| 1 | ID | int | 10 |  |
| 2 | AdminName | Varchar | 255 | NOT NULL |
| 3 | UserName | Varchar | 255 | NOT NULL |
| 4 | FirstName | Varchar | 255 | NOT NULL |
| 5 | LastName | Varchar | 255 | NOT NULL |
| 6 | MobileNumber | Varchar | 255 | NOT NULL |
| 7 | Email | Varchar | 255 | NOT NULL |
| 8 | Status | Varchar | 255 | NOT NULL |
| 9 | Photo | Varchar | 255 | NOT NULL |
| 10 | Password | Varchar | 255 | NOT NULL |
| 11 | AdminRegdate | Varchar | 255 | NOT NULL |

* **Animal Table:-**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SR NO | FIELD NAME | DATA TYPE | SIZE | CONSTAINT |
| 1 | ID | int | 10 | NOT NULL |
| 2 | AnimalName | Varchar | 255 | NOT NULL |
| 3 | CageNumber | Varchar | 255 | NOT NULL |
| 4 | FeedNumber | Varchar | 255 | NOT NULL |
| 5 | Breed | Varchar | 255 | NOT NULL |
| 6 | AnimalImage | Varchar | 255 | NOT NULL |
| 7 | Description | Varchar | 255 | NOT NULL |
| 8 | creationdate | timestamp |  | NOT NULL |

* **Indian Ticket Table: -**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SR NO | FIELD NAME | DATA TYPE | SIZE | CONSTAINT |
| 1 | ID | Int | 10 | NOT NULL |
| 2 | TicketID | Varchar | 100 | NOT NULL |
| 3 | NoAdult | Int | 10 | NOT NULL |
| 4 | Nochildren | Int | 10 | NOT NULL |
| 5 | Adultunitprice | Varchar | 50 | NOT NULL |
| 6 | Childunitprice | Varchar | 50 | NOT NULL |
| 7 | Postingdate | timestamp |  | NOT NULL |

* **Ticket Type Table: -**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SR NO | FIELD NAME | DATA TYPE | SIZE | CONSTAINT |
| 1 | ID | Int | 10 | NOT NULL |
| 2 | Ticket Type | Varchar | 200 | NOT NULL |
| 3 | Price | Varchar | 50 | NOT NULL |
| 4 | creationDate | timestamp |  | NOT NULL |

# **CHAPTER 6**

# **SOFTWARE REQUIREMENT SPECIFICATION**

* **Hardware:**
* Processor: - Core2Duo.
* RAM: - 128MB and above.
* Hard Disk: - 20GB.
* **Software:**
* OS: - Windows.
* Browser: - Google chrome, Mozilla, any JavaScript enabled browser.

**SERVER SIDE**

* **Hardware:**
* Processor: -Intel core i3.
* RAM: - 4 GB.
* Hard Disk: - 500GB.
* **Software:**
* OS: - Windows.
* Xampp Server.
* Notepad ++.
* Visual Studio Code.

.

### CHAPTER 7

### TESTING

### INTRODUCTION

Testing is a process used to help identity the correctness, completeness and quality of developed computer software. With that in mind, testing can never completely establish the correctness of computer software. In other words, Testing is nothing but criticism or comparison. Here comparison in the sense comparing the actual value with expected one.

The quality of the application can normally do vary widely from system to system but some of the common quality attributes include reliability, stability, portability, maintainability and usability.

Testing helps in Verifying and Validating if the software is working as it is intended to be working. Thins involves using Static and Dynamic methodologies to Test the application.

Because of the feasibility of its human designers and its own abstract, complex nature, software development must be accompanied by quality assurance activities. It is not unusual for developers to spend $0% of the total project time on testing. For life- critical software, testing can cost 3 to 5 times as much as all other activities combined. The destructive nature of testing requires that the developer discard preconceived notions of the correctness of his/her developed software.

* **SOFTWARE TESTING FUNDAMENTALS**

The goal of testing is to find errors, and a good test is one that has a high probability of finding an error. Therefore, you should design and implement a computer-based system or a product with -testability in mind. At the same time, the tests themselves must exhibit a set of characteristics that achieve the goal of finding the most errors with a minimum of effort.

**Testability:** James Bach provides the following definition for testability: Software testability is simply how easily [a computer program] can be tested. The following characteristics lead to testable software.

**Operability:** The better it works, the more efficiently it can be tested.11 If a system is designed and implemented with quality in mind, relatively few bugs will block the execution of tests, allowing testing to progress without fits and starts.

**Observability:** What you see is what you test.!! Inputs provided as part of testing produce distinct outputs. System states and variables are visible or quarriable during execution. Incorrect output is easily identified. Internal errors are automatically detected and reported. Source code is accessible. Controllability: The better we can control the software, the more the testing can be automated and optimized. All possible outputs can be generated through some combination of input, and FO formats are consistent and structured. All code is executable through some combination of input. Software and hardware states and variables can be controlled directly by the test engineer. Tests can be conveniently specified, automated, and reproduced.

**Decomposability:** By controlling the scope of testing, we can more quickly isolate problems and perform smarter retesting. The software system is built from independent modules that can be tested independently.

**Simplicity:** The less there is to test, the more quickly we can test it. The program should exhibit functional simplicity (e.g., the feature set is the minimum necessary to meet requirements); structural simplicity (e.g., architecture is modularized to limit the propagation of faults), and code simplicity (e.g., a coding standard is adopted for ease of inspection and maintenance).

**Stability:** The fewer the changes, the fewer the disruptions to testing. Changes to the software are infrequent, controlled when they do occur, and do not invalidate existing tests. The software recovers well from failures

**Understandability**: The more information we have, the smarter we will test. The architectural design and the dependencies between internal, external, and shared components are well understood. Technical documentation is instantly accessible, well organized, specific and detailed, and accurate. Changes to the design are communicated to testers.

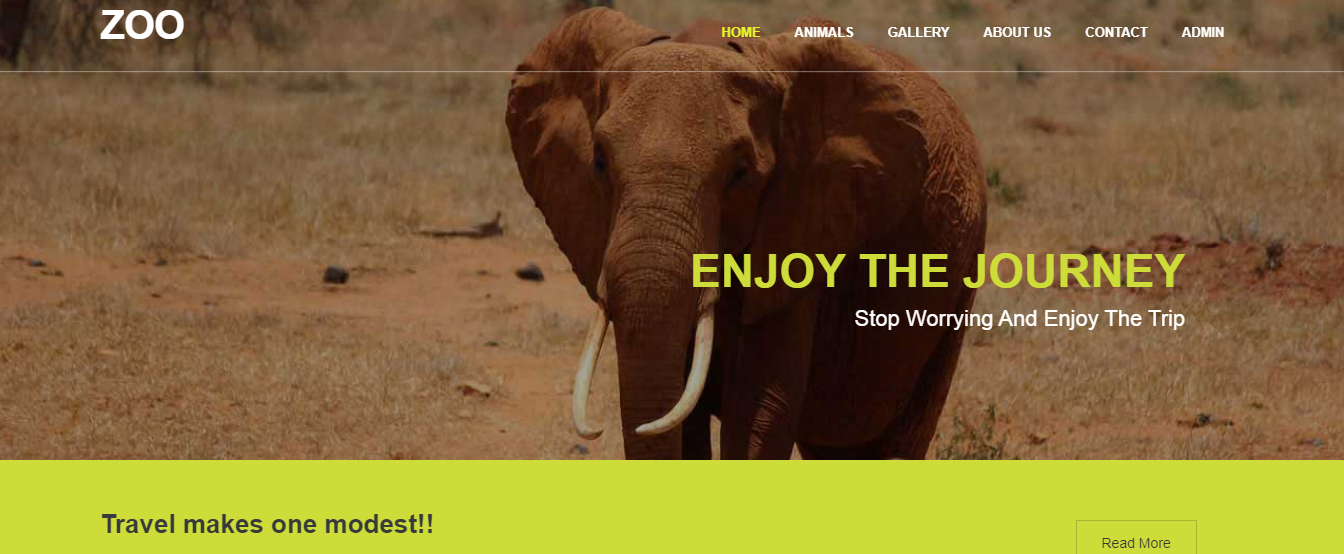
* **GOALS AND OBJECTIVE:**
* The objectives of testing can be stated as:
* To execute the program with the intent of finding errors.
* Writing test cases with the high probabilities of finding an as-yet undiscovered error**.**
* **TEST CASES**

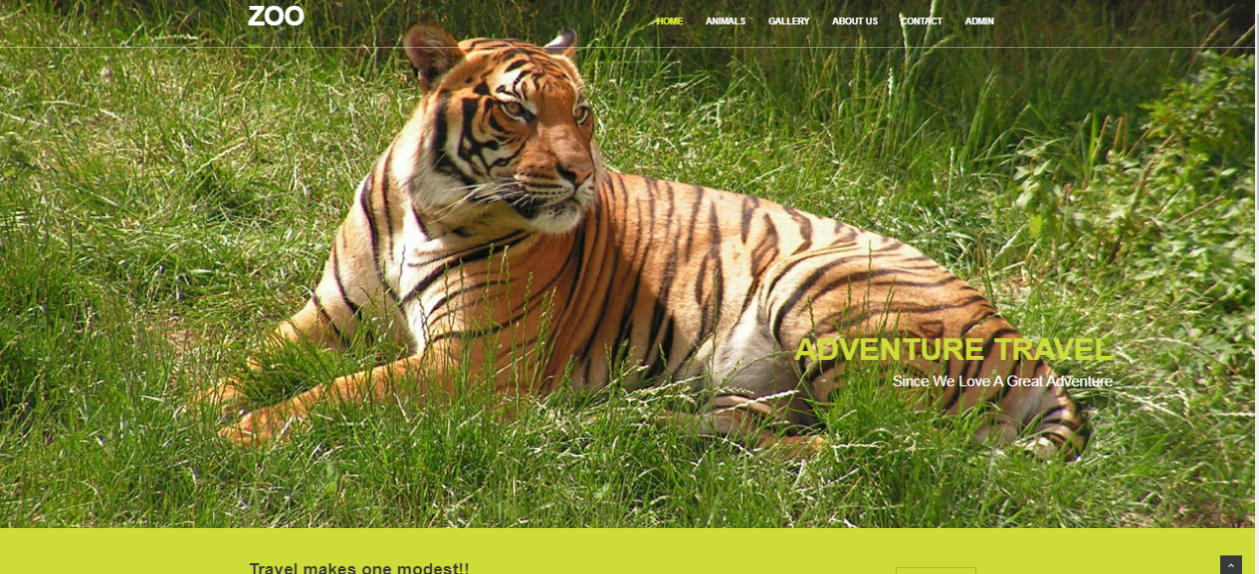
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test case Name** | **Test Case**  **Description** | **Expected**  **Value** | **Actual Value** | **Result** |
| 1 | Functionality | User Interaction Testing Transaction testing | User Interaction Testing Transaction  testing | User Interaction Testing Transaction testing | Pass |
| 2 | Performance | UI  responsiveness Transaction completion time(s) Peak load performance Longevity | UI  responsiveness s Transaction completion time(s) Peak load performance Longevity | UI  responsiveness Transaction completion time(s) Peak load performance Longevity | Pass |
| 3 | Usability | User Experience | User  Experience | User  Experience | Pass |
| 4 | Installation and Provisioning | Installation process Un- installation process User provisioning and de-provisioning | Installation process Un- installation process User provisioning  and de- provisioning | Installation process Un- installation process User provisioning  and de- provisioning | Pass |
| 5 | Admin login | Enter the admin username and password | Admin login successful  Entered in the  Admin Dashboard | Admin login successful Entered in The Admin Dashboard | Pass |

**CHAPTER 8**

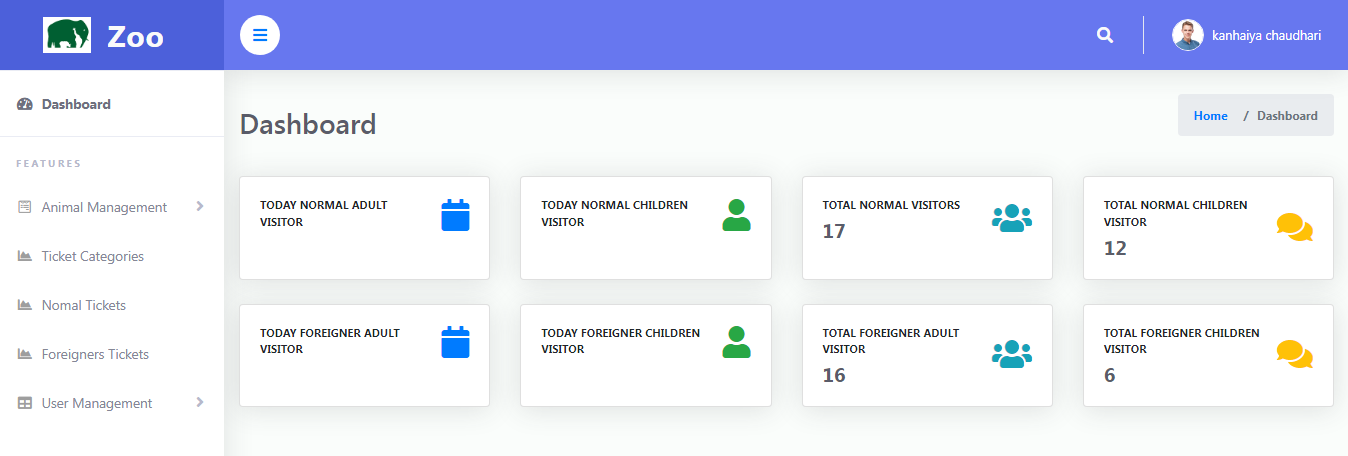
**SNAPSHOTS**

**Home page: -**

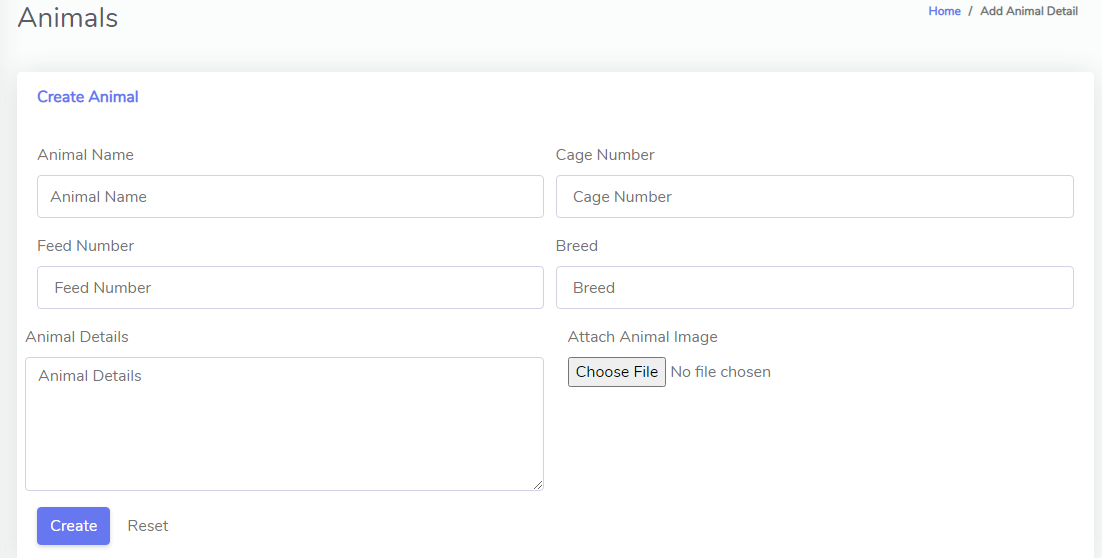
****



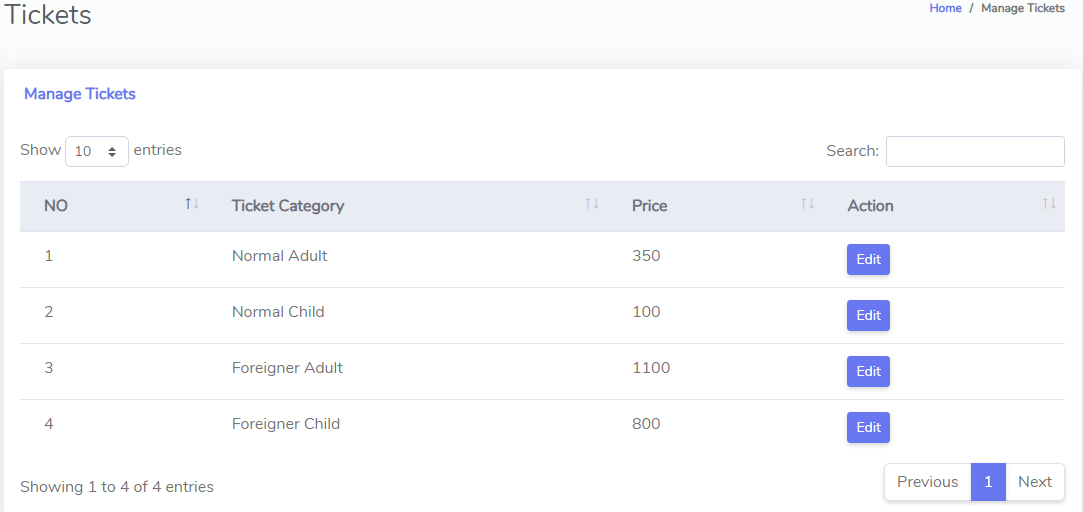
**Admin Dashboard page: -**

****

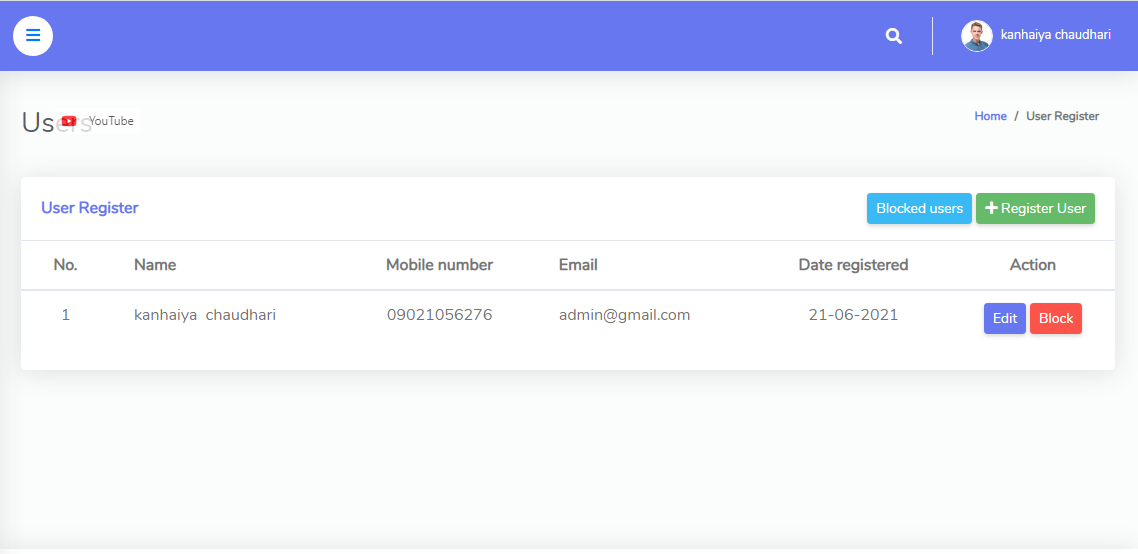
**Animal Registration page: -**



**Ticket Management Page: -**



**Admin Management page: -**



### CHAPTER 9

### ADVANTAGES

* Manage Data of Animal.
* Manage Tickets data.
* Create new User for admin Dashboard.
* Delete new User Existed in Admin dashboard.
* Add Admin.
* Delete Admin.
* There is option to upload image of animal.
* There is no paper work to lost data also saved trees.
* Data about animal quickly find out.
* Also, Data about Animal quickly find out.
* If anyone has a problem they can directly contact with zoo.
* It is helps to grow our tourism industry.

**CHAPTER 10**

**APPLICATIONS**

* Useful in Cow Shed.
* Can use in Zoological gardens.
* Can Use in Botanical Gardens.
* Also use as Government Tourism site.
* Useful in Tiger Reserves.
* Useful in Bird sanctuary.

**11. CHAPTER**

**FUTURE SCOPE**

Scope of any system is inevitable over the period of years. As this project is user friendly, any sort of enhancement which results in quick response to the user provides more details, reduce manual effort would be given a through for future enhancements. Functional Enhancement

**From the functional point of view**

Currently, reports are generated only for quotation module. This can be enhanced to get the reports for all other modules as well

**Technical Scope**

This project has been developed using MySQL as back end technology currently. A newer version is introduced and available. When the newer version of database with more functionality is suitable for this project, current version can be upgraded

Similarly, with the newer version of front-end technology - the visual studio Code is available the new techniques can be used to use this system as more user friendly.

**12. CHAPTER**

**CONCLUSION**

Zoo Management System project is very helpful to maintain the whole zoo it is very easy to handle and helps the environment in saving paper.

As a Security measure, the user who has valid username and password only can log into the system. So, the users cannot use this system without proper authentication. This, in turn, maintains the confidently of the customer details which is quit evidence for customer satisfaction.

Since the project uses the PHP as the front-end program, it is easy for any user with Basic English knowledge and basic computer concepts to use this system with less training. As the project designed with lots of validation for the incorrect user’s input, even if the user enters wrong data, the system itself will prevent the user form entering wrong inputs and proceed further. This helps to maintain valid data in the database.

This project helps the staffs to maintain the animal’s details very efficiently. Also, there cannot be any calculation errors in billing since this system automatically calculates the amount and generates the bills. To conclude, this project is well documented and will be very helpful for others who need guidance to develop their project. This project is developed to consume time and reduce manual labor. Maintenance is also very easy.

**13. CHAPTER**

**REFERENCE**

* **BOOKS:-**

**PHP&MYSQL:** Robin Nixon

**Software Engineering:** Roger Pressman

* **SITES:-**

**PHP:**

[www.phpgurukul.com](http://www.phpgurukul.com)

[www**.**php.in](http://www.php.in)

[www.Coadacadamy.com](http://www.Coadacadamy.com)

**MYSQL:**

[www.w3c.com](http://www.w3c.com)

[www.Mysql.com](http://www.Mysql.com)