R.V. COLLEGE OF ENGINEERING, BENGALURU-560059

(Autonomous Institution Affiliated to VTU, Belagavi)

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



**WILDLIFE SANTURY MANAGEMENTN SYSTEM**

##### Mini - Project Report

###### *Submitted by*

**HEMANTH RAO K N 1RV17CS410**

**UJWALAKAVYA J 1RV17CS435**

***in partial fulfillment for the requirement of 5th Semester***

***DBMS Laboratory Mini Project (12CS54)***

**Under the Guidance of**

**Mrs. SUMA B**

**Assistant Professor**

**CSE, RV college of Engineering**

**Academic Year 2018– 2019**

**R.V. COLLEGE OF ENGINEERING, BENGALURU - 560059**

**(Autonomous Institution Affiliated to VTU, Belagavi)**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

****

**CERTIFICATE**

Certified that the project work titled **‘WILDLIFE SANTURY MANAGEMENT SYSTEM’** is carried out by **HEMANTH RAO K N(1RV17CS410), UJWALAKAVYA J(1RV17CS435)**who are bonafide students of R.V. College of Engineering, Bengaluru, in partial fulfillment of the curriculum requirement of 5th Semester Database Design Laboratory Mini Project during the academic year **2018-2019**. It is certified that all corrections/suggestions indicated for the internal Assessment have been incorporated in the report deposited in the departmental library. The report has been approved as it satisfies the academic requirements in all respect laboratory mini-project work prescribed by the institution.

**Signature of Faculty In-charge Head of the Department**

**Dept. of CSE, RVCE**

**External Examination**

**Name of Examiners Signature with date**

**1**

**2**

Table of Contents

1. Introduction 4

1.1 Objective 4

1.2 Scope 4

2. Software Requirement Specification 5

2.1 Software Requirements. 5

2.2 Hardware Requirements 5

2.3 Functional Requirements 6

3. ER Diagram 8

4. Detailed Design 9

4.1 DFD Level 0 9

4.2 DFD Level 1 10

4.3 DFD Level 1 10

5. Relational Schema 12

6. Normalization 14

7. Conclusion 17

**8. References** **……………………………………………………………………………………18**

**9. Appendix: Snapshot………………………………………………………………………....19**

**1. INTRODUCTION**

The wildlife sanctuary system is a web application to maintain the entire details of the sanctuary which includes the animal, birds, staff and visitors’ details.

Unlike the manual system which is very time consuming and lazy, it is more prone to Errors. The manual system are facing the problems of issuing a ticket and maintaining the animal’s data. It takes more time and difficult to maintain. The data are stored in papers or in MS Excel. This leads to the difficulty of calculating the data, maintaining and storing the hard copy of records are difficult and costly. Even requires more manual work.

The database system uses a new way to overcome the above problem by storing all the data in database that will help to provide the tickets effectively without any corrupted data or information and to provide a secured database to store all animals’ details.by this way, the history record can be maintained without much manual effort and storage of paper. This reduces the usage of papers which in turns helps to keep Green Environment.

**1.1 OBJECTIVE:**

The objective of the system is to serve the smooth administration, efficiently maintain the database and documents of wildlife.The application keeps a backup of all the details maintained related to the sanctuary. The wild life details like their birth, death, date of arrival, food habits, health condition, species to which they belong to are managed.The staff details contains unique id, name, designation, working hours, salary and phone number. The visitors details contains ticketing data, the fare they pay and the donations they make out for the welfare of wild animals.

The system have 3 users namely the admin, staff and the visitor. The authorized admin registers the staff and only the registered staff can login and access the data. The wildlife’s details updating and the assignment of the job to the staff are done by admin of the system. The visitors ticked booking and donations made by them are managed by staff members. Even the food and health conditions of animals are monitored by staff. The visitors can view the animal details and contact information of the sanctuary.

**1.2 SCOPE:**

The scope of the project is to give a simple, attractive and user friendly system to simplify the work as well as to reduce the effort while doing work with offline or in old method. The system standardize the collection of wildlife sanctuary Information in a single database.

Secure registration and profile management facilities for registered members are provided and can track Species Profiles, Add new animal details and update the existing data. Workers details and offline ticket issuing for visitors details are managed efficiently. Even the visitor’s data and donations made by them are managed. User and staff auditing of wild animals information are provided.

**2. SOFTWARE REQUIREMENT SPECIFICATION**

      Software Requirements Specification (SRS) is a primary document for development of a Software. It is written by Business Analysts who interact with client and gathers the requirements to build the software. It goes through several reviews throughout the software life cycle. The software requirements specification lays out [functional](https://en.wikipedia.org/wiki/Functional_requirement) and [non-functional requirements](https://en.wikipedia.org/wiki/Non-functional_requirements), and it may include a set of [use cases](https://en.wikipedia.org/wiki/Use_case) that describe user interactions that the software must provide.

It establishes the basis for an agreement between customers and contractors or suppliers on how the software product should function. Software requirements specification is a rigorous assessment of requirements before the more specific system design stages, and its goal is to reduce later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules.

**2.1 Software Requirements**

* The system is developed in Java programming language.

JAVA is an object-oriented programming language with a built-in application programming interface (API) that can handle graphics and user interfaces and it is platform independent.

* HTML and CSS is used for graphical user interface.
* MySQL and MongoDB is used as backend for storing all the data.

MySQL is an open-source relational database management system, MySQL is written in [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B). Its SQL parser is written in [yacc](https://en.wikipedia.org/wiki/Yacc), but it uses a home-brewed [lexical analyzer](https://en.wikipedia.org/wiki/Lexical_analysis). It provides an efficient way to manage structured data.

MongoDB is a free and open-source cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schemata and makes maintaining unstructured data easy.

* The system requires Apache Tomcat Apache, it is used to deploy Java Servlets and JSPs. So in Java project one can build WAR (Web ARchive) file, and just drop it in the deploy directory in Tomcat. So basically Apache is an HTTP Server, serving HTTP. Tomcat is a Servlet and JSP Server serving Java technologies.

**2.2 Hardware Requirements**

To provide better performance the system is built on the following hardware requirements. A system with minimum of 700 megabytes of RAM and Windows 7 operating system. Intel Core duo Processor with speed 2.0 GHZ or more and hard disk of 80 GB or more.

**2.3 Functional Requirements**

**Functional requirement** defines a function of a [system](https://en.wikipedia.org/wiki/System) or its component, where a function is described as a specification of behaviour between outputs and inputs.

Functional requirements may involve calculations, technical details, data manipulation, processing, and other specific functionality that define what a system is supposed to accomplish.

## Login and registration

This section involves the login and registration process with have the high priority as maintaining the security of the database is very important. Through login process only the authorized authenticated users can access the system data.

2.3.1.1 Stimulus/Response Sequences

Stimulus: 1 User clicks on login button to launch the login page.

Response: The system will load a login window.

Stimulus: 2 User has to entry the username and password.

Response: The system will authenticate the user by comparing the entered data with the database data and if the user is valid application enters into user profile, else an invalid username and password message is displayed.

Stimulus: 3 Admin clicks on registration button

Response: The system will load the registration form

Stimulus: 4 the admin enters all details of the staff and submits

Response: The registered user data will be stored in the database

2.3.1.2 Functional Requirements

REQ-1: Form - The system should provide forms for login and registration.

REQ-2: Type Selection - The tool shall allow the selection of type of user for login (Admin/Staff)

## Add Animal Details

This section describes the processes involved with the creation of new wildlife habitat feature record. Users will choose the animal type, species and related habitat information. This section have high priority has the application is mainly about managing wildlife data.

2.3.2.1 Stimulus/Response Sequences

Stimulus: 1 User clicks on ADD button to launch add animal details page.

Response: The system will load add animal details page.

Stimulus: 2 User has to entry the details of animal.

Response: The system will store all the data in the database.

Stimulus: 3 Admin clicks on back button

Response: The system returns to the menu page

2.3.2.2 Functional Requirements

REQ-1: Type Selection - The tool should provide a dropdown list to select the type of animal to which it belong to.

REQ-2: Form - The system should provide forms to enter the details of animal.

## Update Animal Details

This section describes the processes involved with the updating the wildlife habitat feature record which is already there in the system .Users will choose the animal type, species and related habitat information. This section have medium priority.

2.3.3.1 Stimulus/Response Sequences

Stimulus: 1 User clicks on update button to launch update animal details page.

Response: The system will load update animal details page.

Stimulus: 2 User can make the changed to the animal database.

Response: The system will store the updated data in the database.

Stimulus: 3 Admin clicks on back button

Response: The system returns to the menu page

2.3.3.2 Functional Requirements

REQ-1: Type Selection - The tool should provide a dropdown list to select the type of animal to which it belong to.

REQ-2: Form - The system load the forms to update the details of animal.

## Ticketing

This section involves issuing tickets to visitors, maintaining their details and donations made by them for the welfare of the wildlife sanctuary.

2.3.4.1 Stimulus/Response Sequences

Stimulus: 1 Staff clicks on ticketing button to launch ticket booking page.

Response: The system will load ticket booking page.

Stimulus: 2 User has to entry the details of visitors and bill.

Response: The system will store all the data in the database after successful payment.

Stimulus: 3 Admin clicks on back button

Response: The system returns to the menu page

2.3.4.2 Functional Requirements

REQ-1: Form -The system should provide forms to enter the details of visitors.

**3. E-R Diagram**

An entity-relationship diagram (ERD) is a data modelling technique that graphically illustrates an information system’s entities and the relationships between those entities. An ERD is a conceptual and representational model of data used to represent the entity framework infrastructure.

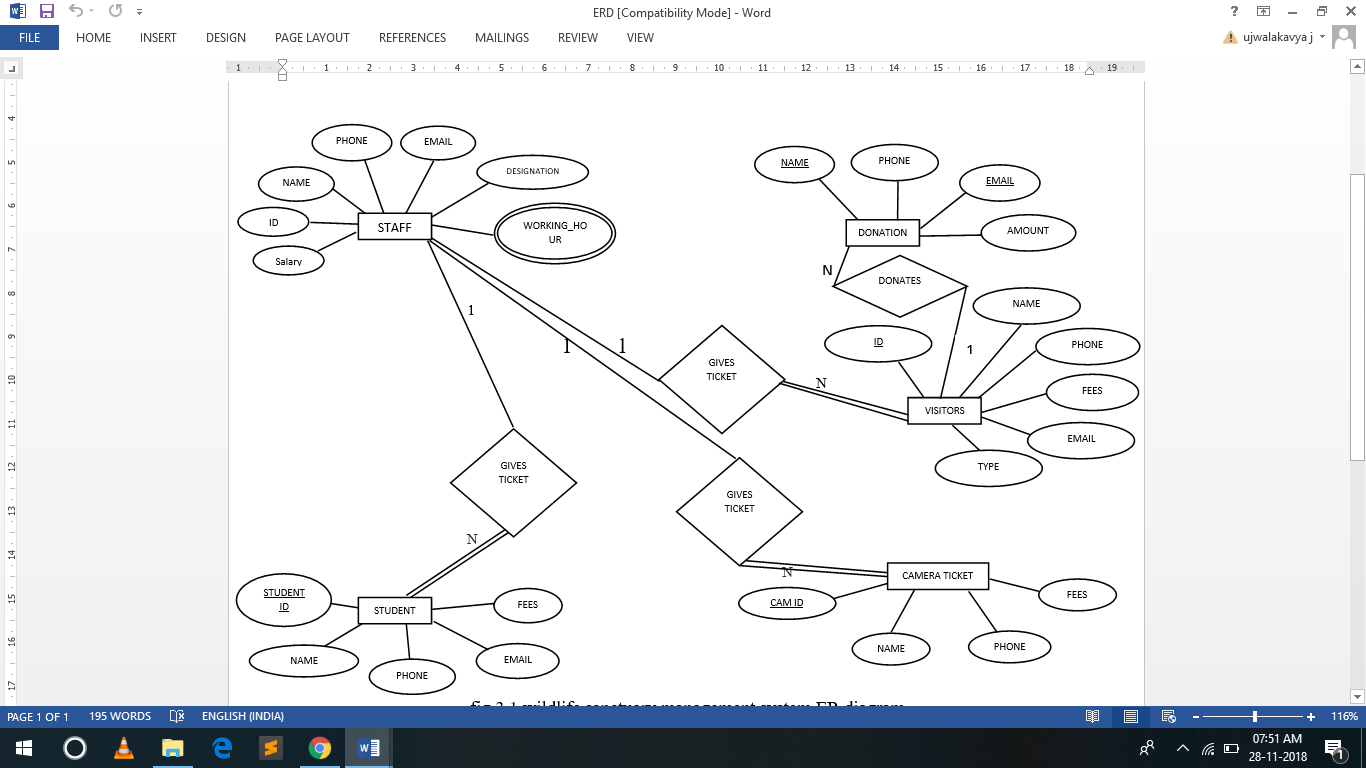


fig 3.1 wildlife sanctuary management system ER diagram

In fig 3.1 there are five entities namely Staff, Student, Visitor, Donation, camera\_ticket and four relationships. There are three separate “Gives\_Ticket” relationship in diagram, because the ticket fare varies for different type of visitors (adult, child and student) and electronic gadgets. The staff will issue tickets for visitors, student and their camera. Visitors may provide donations, single user can provide donation any number of times. A staff handles many users and issues tickets to many visitors and students as defined by “Gives\_Ticket” relationship. The camera must belong to any of the user, either visitor or student.

**4. DETAILED DESIGN**

DFD is a designing tool used in the top-down approach to Systems Design. The context-level DFD is next "exploded", to produce a Level 1 DFD that shows some of the detail of the system being modeled. The Level 1 DFD shows how the system is divided into sub-systems (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole. Level 2 defines the actual data flow and data storage of the system.

**4.1 DFD Level 0**

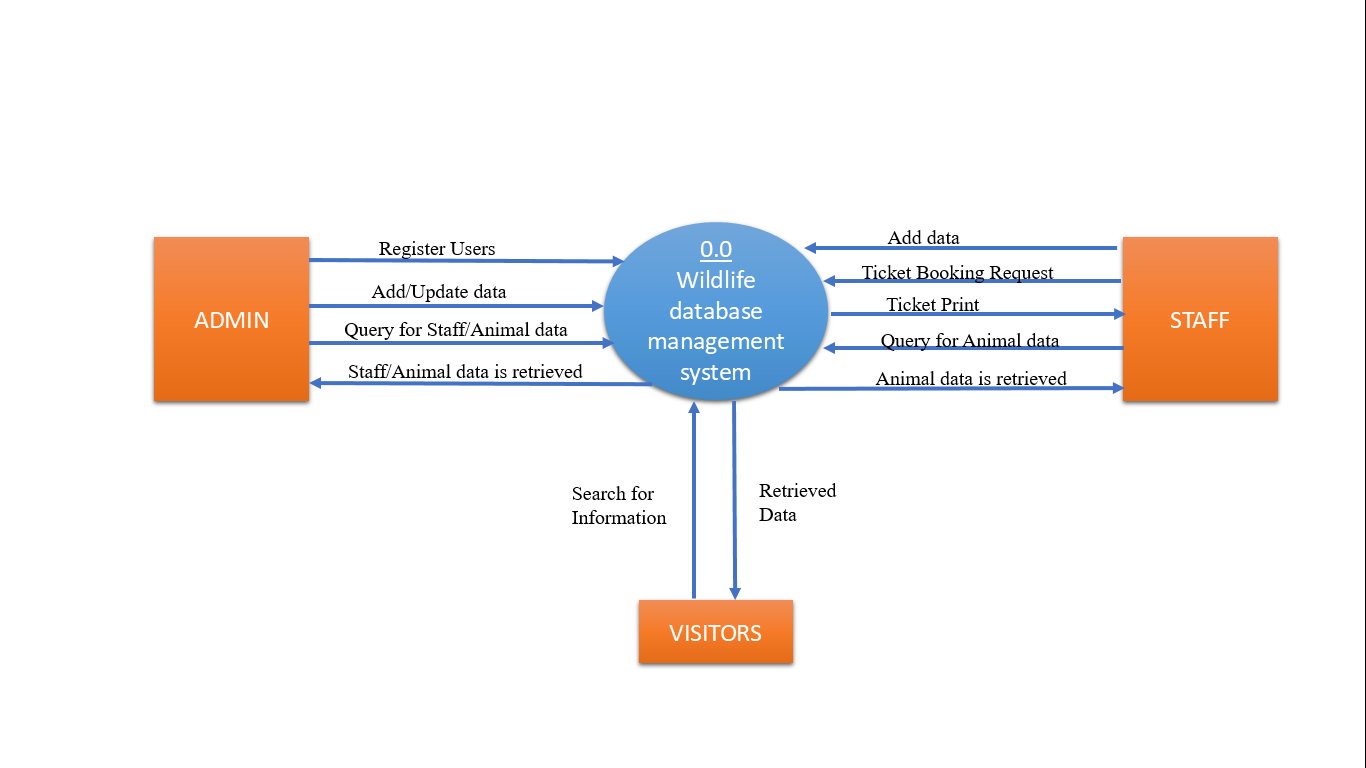


Fig 4.1 level 0 conceptual level diagram

Conceptual level diagram gives an overview of working of the system. It shows actors (admin, staff, visior) and their interface with the system, ie. actions they perform on the system and data flow.

**4.2 DFD Level 1**

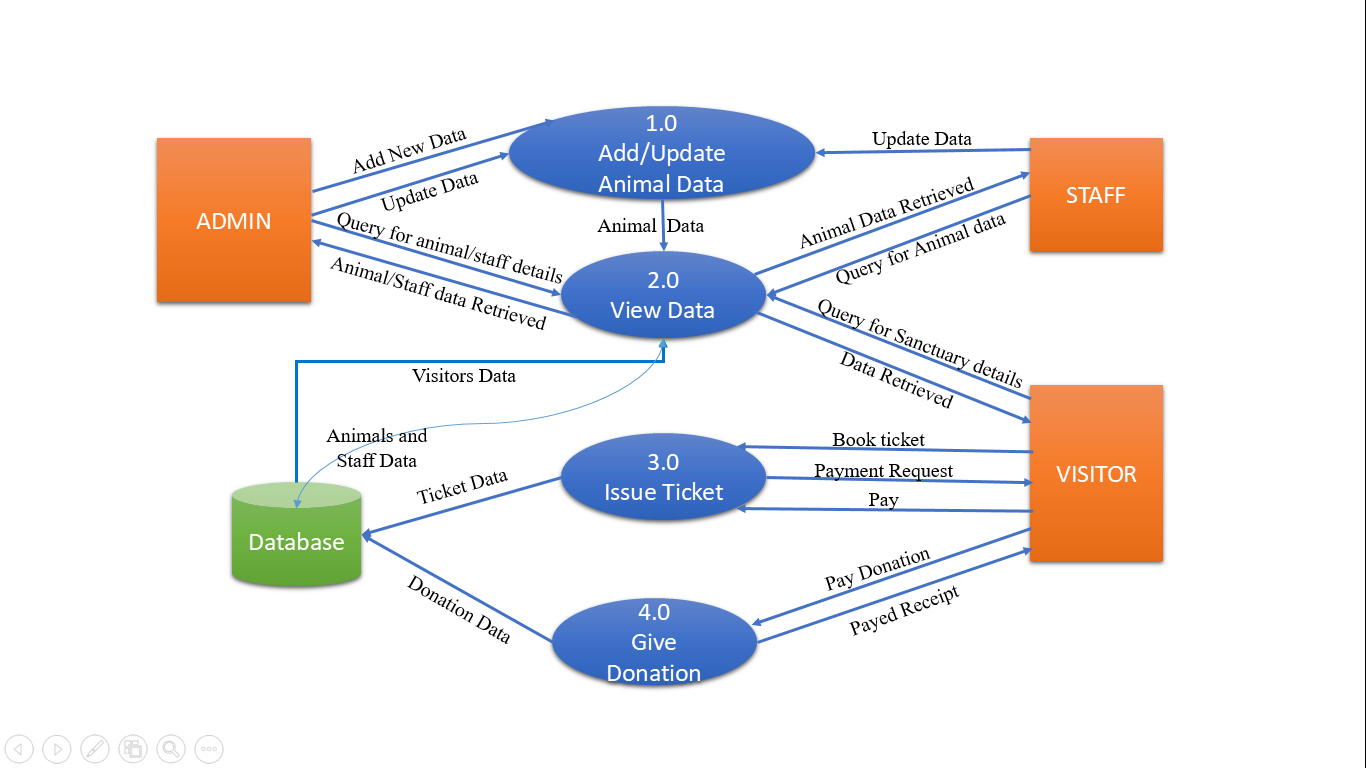


Fig 4.2 level 1 Detail design(contatins all logical Subsystem )

Level 1 data flow diagram shows all the logical sub modules involved in the system, their interface with other module and data flow between modules and the database.

**4.3 DFD Level 2**

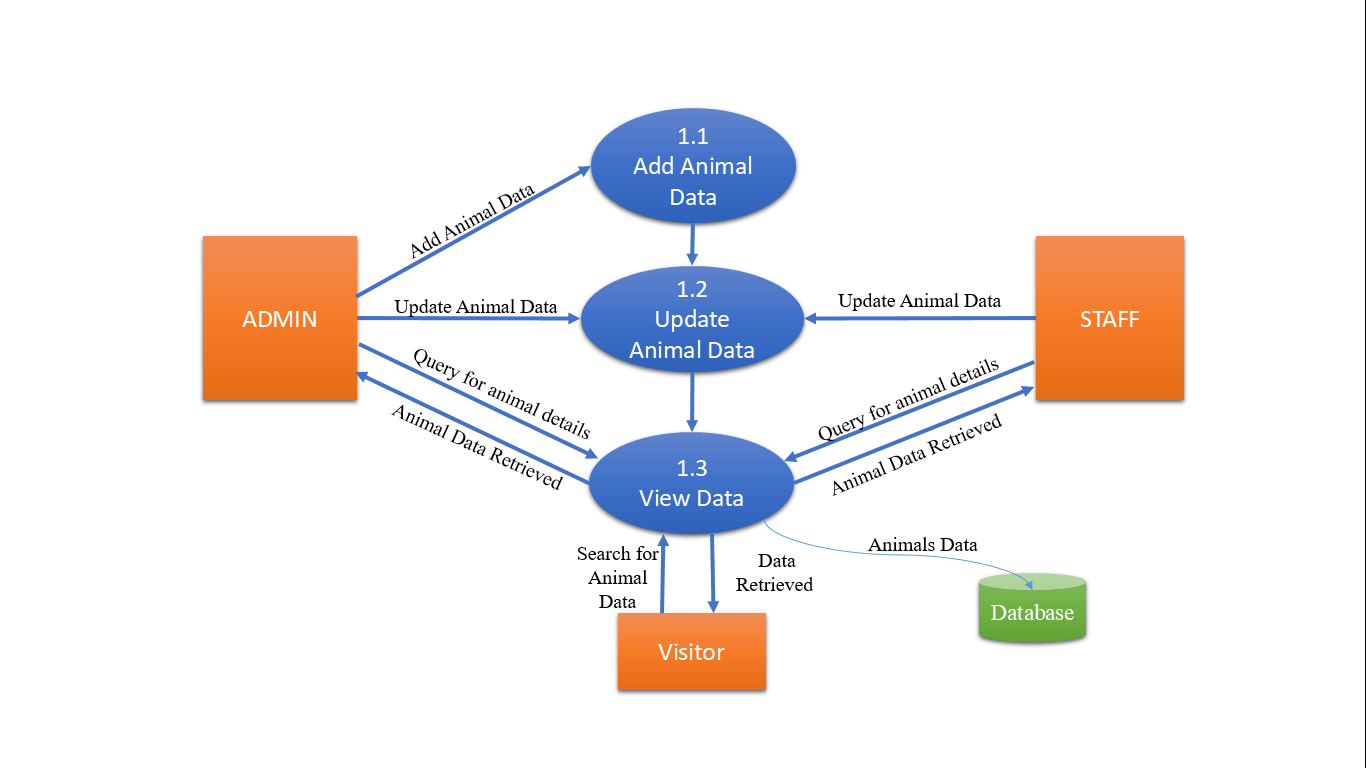


Fig 4.3 level 2 logical working of add/update Animal Data module

Level 2 figure 4.3 shows the further subdivision of the animal module in figure 4.2 . It is divided into add animal data module and update animal data modul

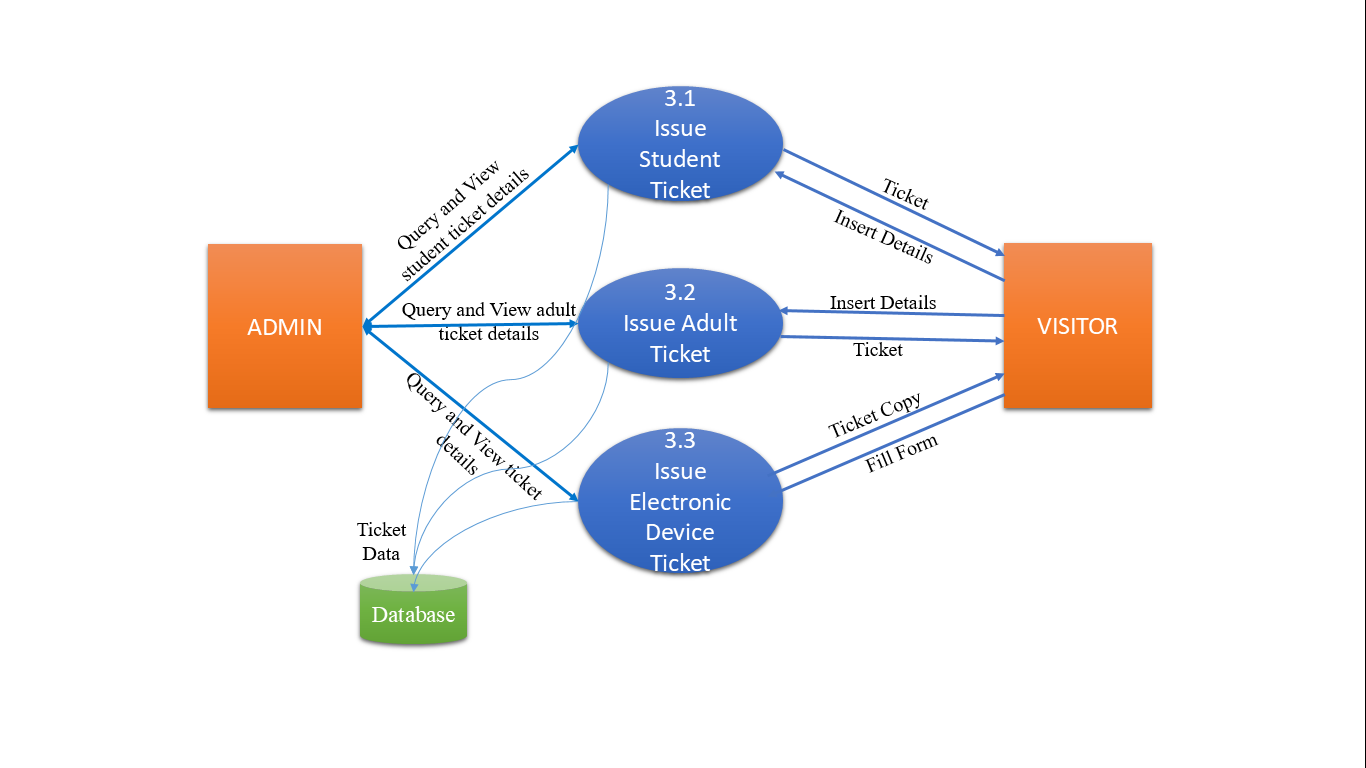


Fig 4.4 level 2 logical working of Issue Ticket module

Level 2 figure 4.4 shows further subdivision of the issue tickets module in figure 4.2. It is divided into student ticket, adult ticket and electronic device ticket modules.

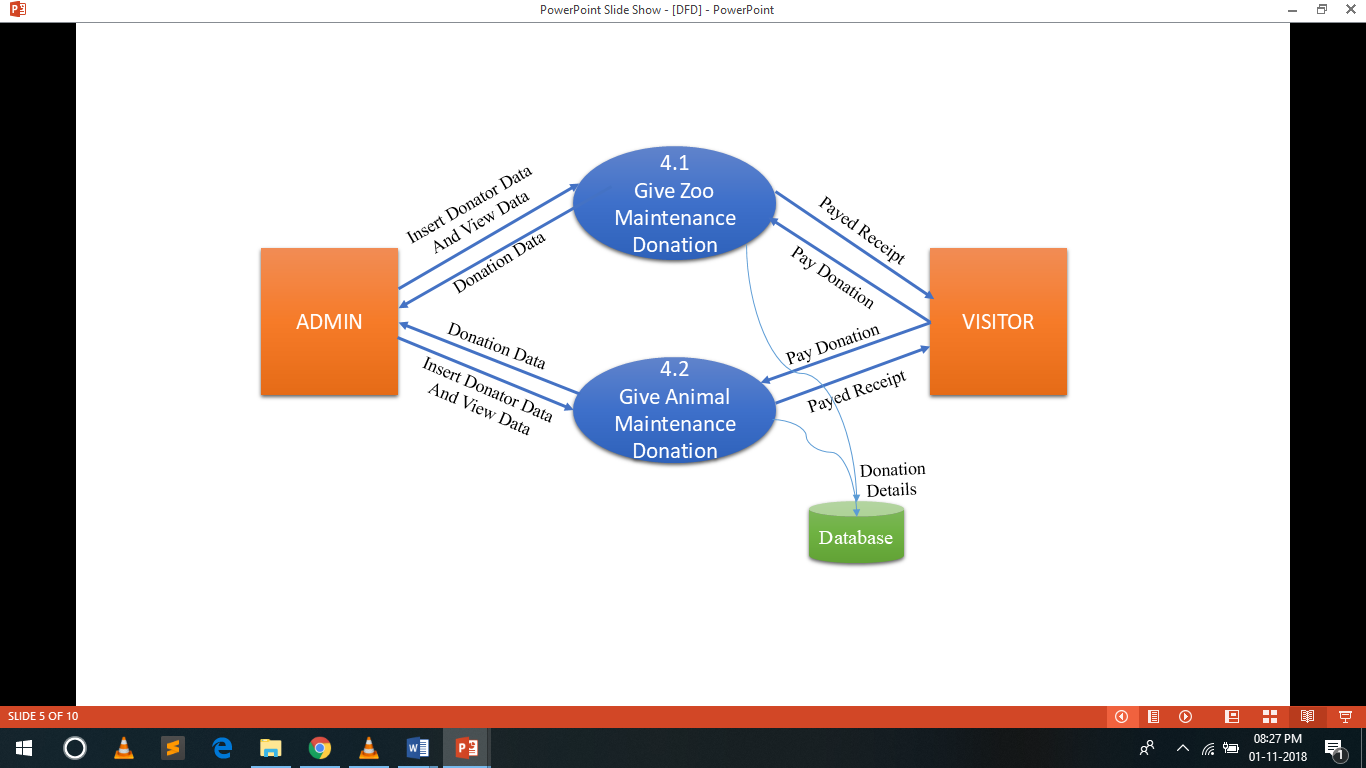


Fig 4.5 level 2 Logical working of Give Donation module

Level 2 figure 4.5 shows futher subdivision of the Give donation module in figure 4.2. It is divided into give zoo maintenance donation and give animal maintenance donation modules.

**5. RELATIONAL MAPPING**

The relational model (RM) for database management is an approach to managing data using a structure and language consistent with first-order predicate logic, where all data is represented in terms of tuples, grouped into relations.

The purpose of the relational model is to provide a declarative method for specifying data and queries: users directly state what information the database contains and what information they want from it, and let the database management system software take care of describing data structures for storing the data and retrieval procedures for answering queries.

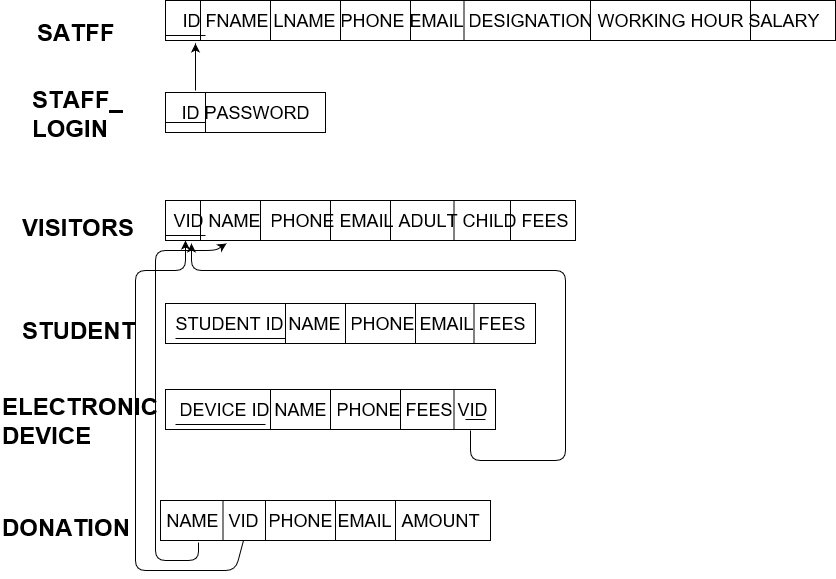


Fig 5.1 Relational mapping of Wildlife Sanctuary Database Management System

After designing the ER diagram of system, we need to convert it to Relational models which can directly be implemented by any RDBMS like Oracle, MySQL etc. The algorithm has the following steps:

**Step 1:** Mapping of Regular Entity Type

In above relations staff.ID, VID, Student ID and Device ID are primary key. The foreign key attributes name and VID of Donation relation, VID of electronic device relation and ID of staff\_login are mapped to corresponding key attributes.

**Step 2:** Mapping of Week Entity Types

Weak entity in Donation relation is mapped to its owner entity.

**Step 3:** Mapping of Binary 1:1 Relationship Types

1:1 relationship between Staff and staff\_login is mapped

**Step 4:** Mapping od Binary 1:N Relationship Types

1:N relationship between Visitor and Donation , Visitor and Electronic device is mapped

**Step 5:** Mapping of Binary M;N Relationship Types

There is no M:N relationship between the relations in wildlife schema

**Step 6:** Mapping of Multivalued Attributes

**Step 7:** Mapping of N-ary Relationship Types

The wildlife does not contain any N-ary relationship types.

**6. NORMALIZATION FORMS (1NF, 2NF, 3NF)**

**First Normal Form (1NF)**

1NF disallows multivalued attributes, composite attributes and their combinations. It disallows relation within relation or relations as attribute values within tuples. The only attribute values permitted in 1NF are single atomic (individual) values.

**Second Normal Form (2NF)**

A relational schema R is in 2NF if every nonprime attribute A in R is fully functional dependent on the primary key of R.

A functional dependency X->Y is a **fully functional dependency** if removal of any attribute A from X means that the dependency does not hold any more.

**Third Normal form (3NF)**

A relational schema R is in 3NF if it satisfies 2NF and no nonprime attribute of R is transitively dependent on the primary key.

**6.1 Staff**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | NAME | PH\_NO | EMAIL | DESIGNATION | WORK\_HR | SALARY |

Fig 6.1(a)

Staff relational schema is in 1NF as all the attributes are single atomic (individual) values and has no composite or multivalued attributes.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | NAME | PH\_NO | EMAIL | DESIGNATION | WORK\_HR | SALARY |

Fig 6.1(b)

Staff relational schema is in 2NF as every nonprime attribute is fully functional dependent on prime attributes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | NAME | PH\_NO | EMAIL | DESIGNATION |

|  |  |  |
| --- | --- | --- |
| DESIGNATION | WORK\_HR | SALARY |

Fig 6.1(c)

Staff relational schema is in 3NF as it satisfies 2NF and no nonprime attribute is transitively dependent on the primary key.

**6.2 Donation**

|  |  |  |  |
| --- | --- | --- | --- |
| NAME | PH\_NO | EMAIL | AMOUNT |

Fig 6.2(a)

Donation relational schema is in 1NF as all the attributes are single atomic (individual) values and has no composite or multivalued attributes.

|  |  |  |  |
| --- | --- | --- | --- |
| NAME | PH\_NO | EMAIL | AMOUNT |

Fig 6.2(b)

|  |  |
| --- | --- |
| NAME | PH\_NO |

|  |  |  |
| --- | --- | --- |
| NAME | EMAIL | AMOUNT |

Fig 6.2(c)

Donation relational schema is in 2NF and 3NF as every nonprime attribute is fully functional dependent on prime attributes and no nonprime attribute is transitively dependent on the primary key.

**6.3 Visitor**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | NAME | PH\_NO | EMAIL | TYPE | FARE |

Fig 6.3(a)

Visitor relational schema is in 1NF as all the attributes are single atomic (individual) values and has no composite or multivalued attributes.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | NAME | PH\_NO | EMAIL | TYPE | FARE |

Fig 6.3(b)

|  |  |  |  |
| --- | --- | --- | --- |
| ID | NAME | PH\_NO | EMAIL |

|  |  |  |
| --- | --- | --- |
| ID | TYPE | FARE |

Fig 6.3(c)

Visitor relational schema is in 2NF and 3NF as every nonprime attribute is fully functional dependent on prime attributes and no nonprime attribute is transitively dependent on the primary key.

**6.4 Student**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STUDENT\_ID | NAME | PH\_NO | EMAIL | FARE |

Fig 6.4(a)

Student relational schema is in 1NF as all the attributes are single atomic (individual) values and has no composite or multivalued attributes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STUDENT\_ID | NAME | PH\_NO | EMAIL | FARE |

Fig 6.4(b)

Student relational schema is in 2NF and 3NF as every nonprime attribute is fully functional dependent on prime attributes and no nonprime attribute is transitively dependent on the primary key.

**6.5 Camera-Ticket**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CAM\_ID | VISITOR\_ID | NAME | PH\_NO | FARE |

Fig 6.5(a)

Camera-Ticket relational schema is in 1NF as all the attributes are single atomic (individual) values and has no composite or multivalued attributes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CAM\_ID | VISITOR\_ID | NAME | PH\_NO | FARE |

Fig 6.5(b)

|  |  |  |  |
| --- | --- | --- | --- |
| CAM\_ID | VISITOR\_ID | NAME | PH\_NO |

|  |  |
| --- | --- |
| CAM\_ID | FARE |

Fig 6.5(c)

Camera-Ticket relational schema is in 2NF and 3NF as every nonprime attribute is fully functional dependent on prime attributes and no nonprime attribute is transitively dependent on the primary key.

**7. CONCLUSION**

In this dissertation we have studied different management systems used in sanctuaries and presented an efficient database management system with the help of that study. Thus we have successfully implemented Wildlife sanctuary management system which helps in administrating the data used for managing the tasks performed in wild life sanctuary.

The system includes both SQL and NOSql database to fulfill different type of data storage needs, attractive user friendly website is created and designed using HTML and CSS.

Enhancement 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.In Future implementation online ticket booking for safari, Zoo visiting and Boating feature can be included in application. Currently only HTML and CSS are used to create user interface, further with the newer version of front end technology - the spring boot, angular JAVA , JAVA Fx are available the new techniques can be used to make this system more user friendly.

**8. REFERENCES**

* <https://ieeexplore.ieee.org/document/7194291>
* <http://www.professionalcipher.com2017/10/dbms-mini-project-wildlife-sanctuary.html>
* <http://iips.icci.edu.iq/images/exam/databases-ramaz.pdf>
* <http://pepa.holla.cz/wp-content/uploads/2016/11/MongoDB-in-Action-2nd-Edition.pdf>

**Tutorial:**

* <https://www.javatpoint.com/java-tutorial>
* <https://www.w3schools.com/>
* [www.mysqltutorial.org/](http://www.mysqltutorial.org/)
* <https://www.tutorialspoint.com/mongodb/>
* <https://www.tutorialspoint.com/eclipse/>

**9. APPENDIX: SNAPSHOTS**

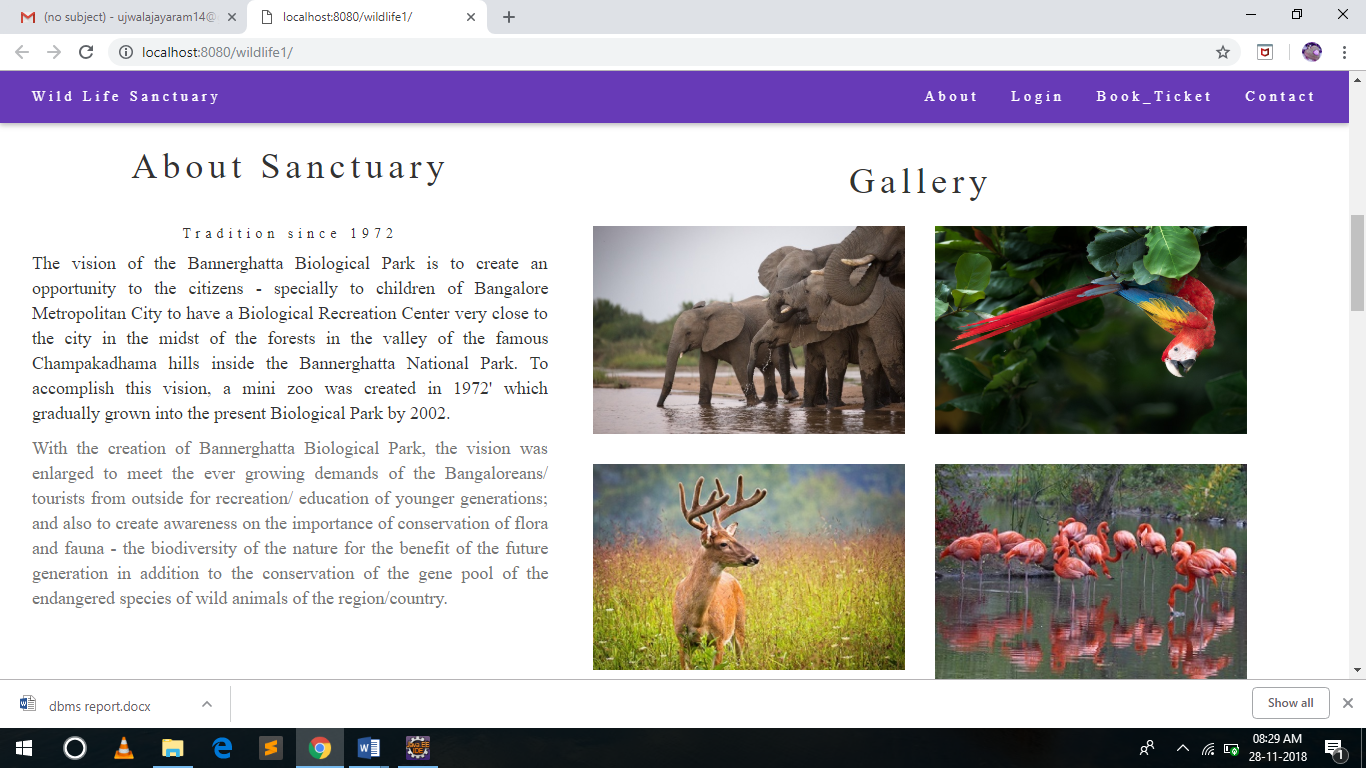


Fig. 9.1 Front Page

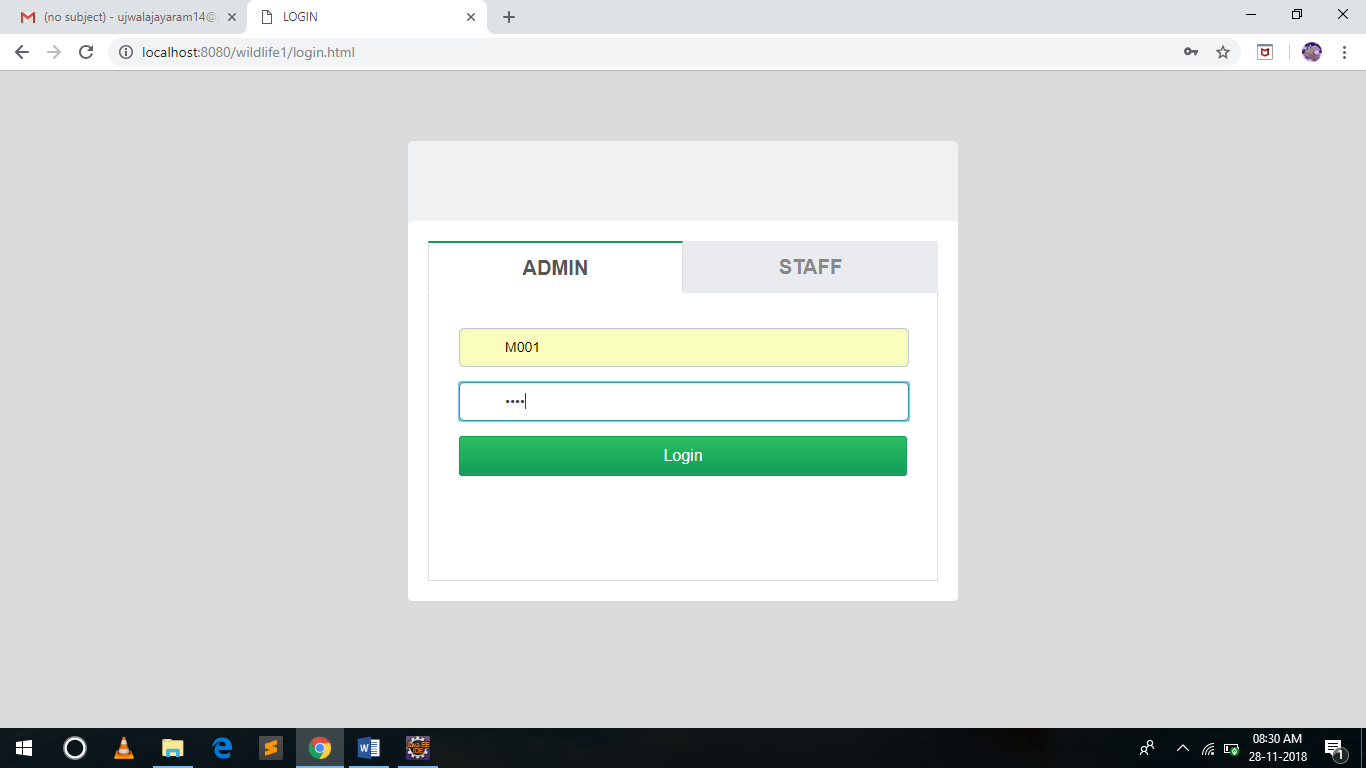


Fig. 9.2 LoginPage

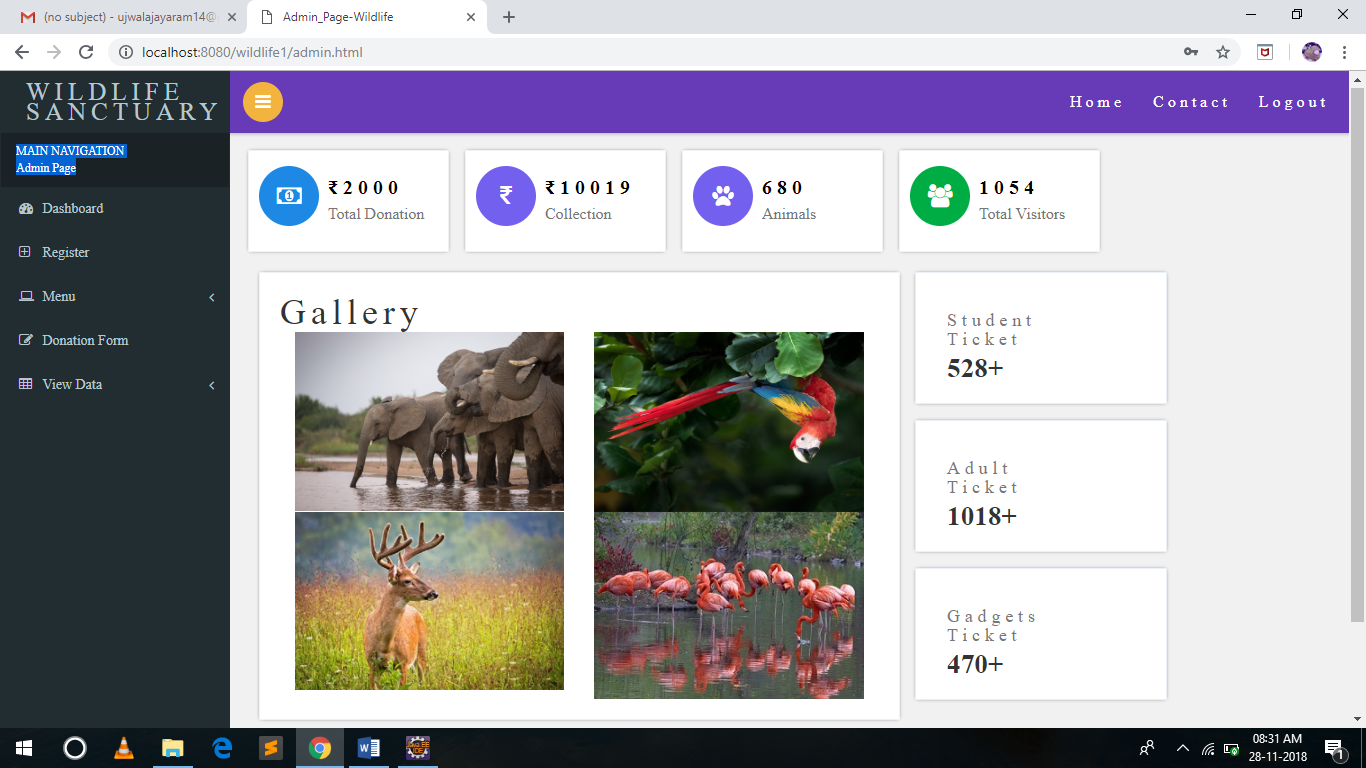


Fig. 9.3 Admin Dashboard

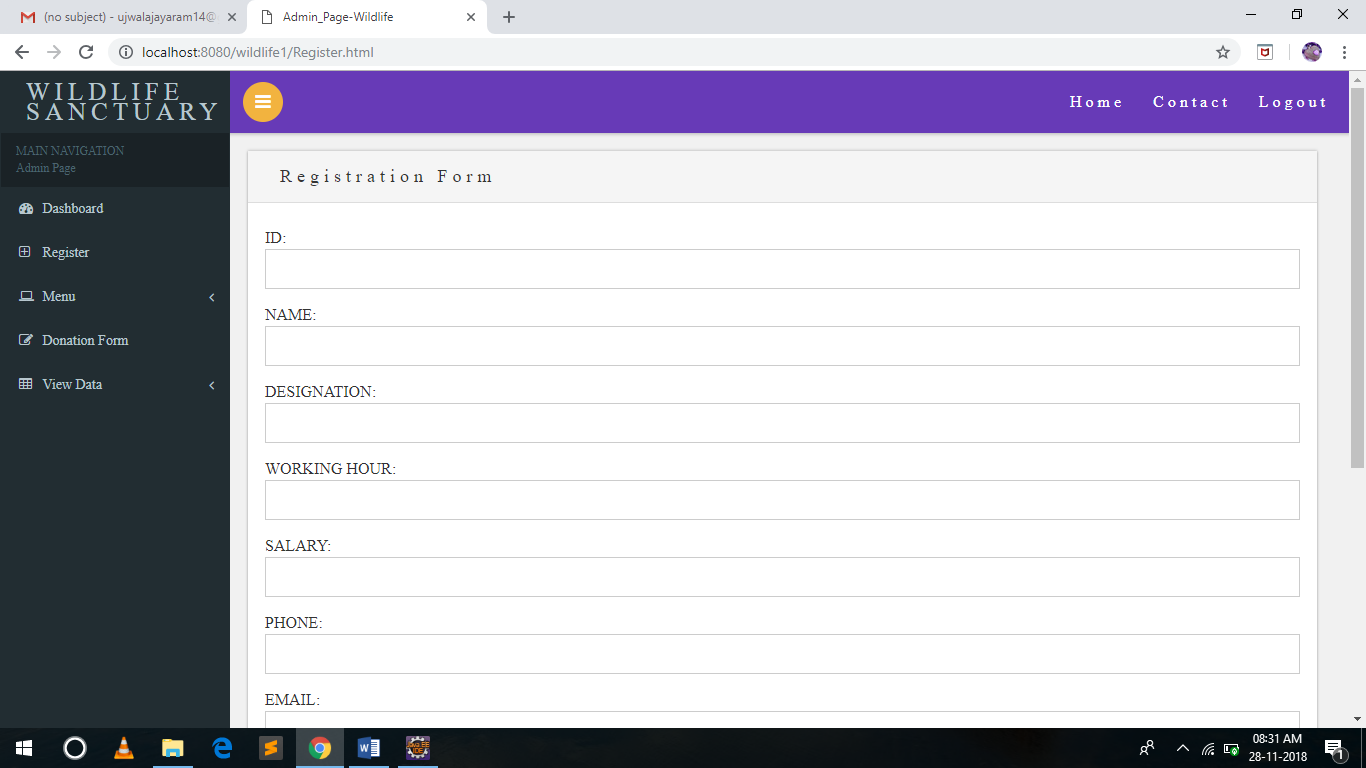


Fig. 9.4 Staff Registration Form

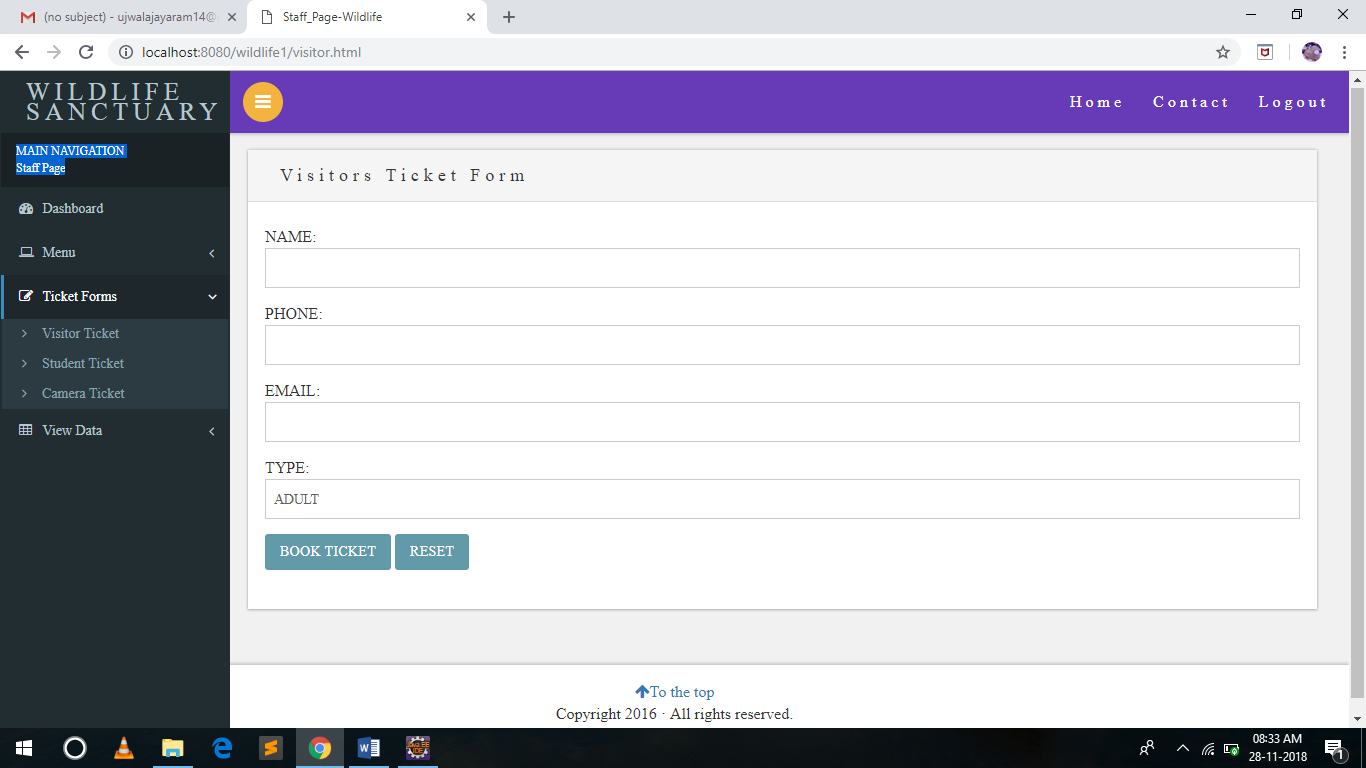


Fig. 9.5 Ticket Booking for visitors (Staff Dashboard)

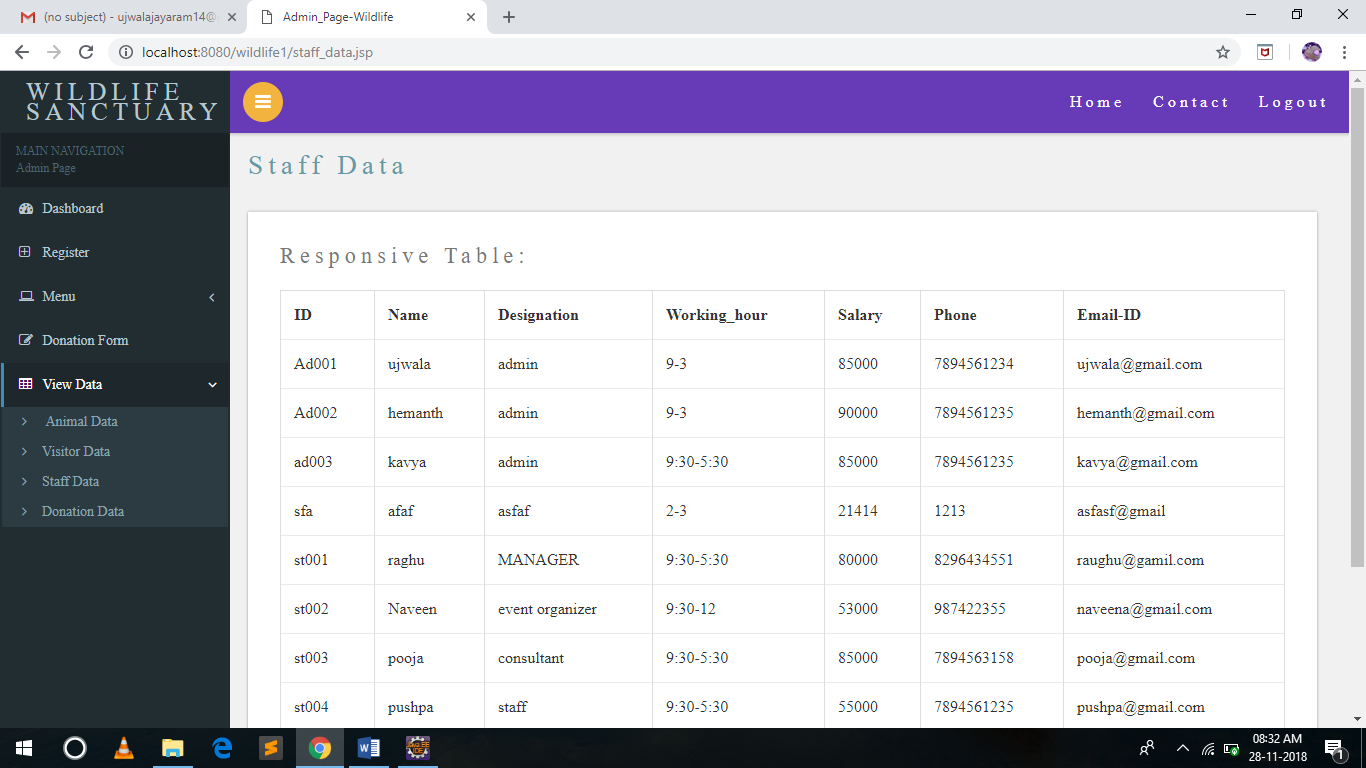


Fig. 9.6 List of Staffs