# KARNATAK UNIVERSITY

# DHARWAD

Janata Shikshana Samiti's

## SHRI MANJUNATHESHWARA INSTITUTE OF UG & PG STUDIES, VIDYAGIRI, DHARWAD - 580 004

A

PROJECT REPORT ON

# "Bird Sanctuary System"

BACHELOR OF COMPUTER SCIENCE (BSc (CS))

OF

KARNATAK UNIVERSITY, DHARWAD

PROJECT GUIDED BY:

Smt. Soumya Singh

Submitted by:

| | |
|---|---|
| Kavya Hegde | Vinoda Patil |
| BSc (CS) VI Semester | BSc (CS) VI Semester |
| 20M10132 | 20M10186 |

DEPARTMENT OF COMPUTER SCIENCE 2022-23

**Janata Shikshana Samiti's**

**Shri Manjunatheshwara Institute of UG & PG Studies,**
**Vidyagiri, Dharwad. 580004**

# CERTIFICATE

*This is to certify that Ms. Kavya Hegde and Ms. Vinoda Patil has satisfactorily completed Project Work entitled "Bird Sanctuary System" for the partial fulfillment of BSc (CS) prescribed by Karnatak University Dharwad during the academic year 2022-2023.*

| | | |
|---|---|---|
| Smt. Soumya Singh | Prof. Vivek. M. Laxmeshwar | Dr. Ajith Prasad |
| [Project Guide] | [Head of The Department] | [Principal] |

Examination Register No: Examiners

1. 20M10132, Kavya Hegde 1. ................................

2. 20M10186, Vinoda Patil 2. ................................

# ACKNOWLEDGEMENT

*The successful presentation of this project is acknowledgements of the immense support extended by JSS SMI UG & PG STUDIES, DHARWAD, which has provided us opportunity to fulfill the most cherished & desired way to reach our goal.*

*We would like to express our heartfelt thanks to our President Shri Shri Vishwaprasanna Teertha Swamiji of Sri Pejavarmath, Udupi and Parama Pujya Dr. D. Veerendra Heggadeji, the Chairman of Janata Shikshana Samiti & Dharmadhikari of Shree Kshetra Dharmasthala.*

*We would like to express our sincere and hearty thanks of gratitude to our beloved Secretary of JSS & Principal Dr. Ajith Prasad and our Head of the Computer Science Department Prof.Vivek.M. Laxmeshwar who gave us the golden opportunity to do this wonderful project on the topic "**Bird Sanctuary System**", which also helped us in doing a lot of research and exposed to lot of new information that would help us in our mere future.*

*We would also take this opportunity to offer our sincere gratitude to our Project guide Smt. Soumya Singh for his excellent support throughout the development of this project and providing the necessary information on our demand at all the times.*

***Kavya Hegde & Vinoda Patil***

# DECLARATION

*We, **Kavya. B. Hegde & Vinoda. M. Patil** students of Sixth Semester BSc (CS), Department of Computer Science, JSS SHREE MANJUNATHESHWARA INSTITUTE OF UG & PG STUDIES, VIDYAGIRI, DHARWAD affiliated to Karnatak University, Dharwad hereby declare that the Project entitled **"Bird Sanctuary System"** submitted in partial fulfillment of the course requirement for the award of degree in Bachelor of Computer Science, Karnatak University, Dharwad during the academic year 2022-2023. We have not submitted the matter embodied to any other University or Institution for the award of any other degree.*

***Date:*** ***Kavya. B. Hegde***

***Place: Dharwad*** ***Vinoda. M. Patil***

# CONTENTS

# THE PROJECT SYNOPSIS

## INTRODUCTION:

India is home to vibrant animal and bird diversity; Birds have ecological value as important elements of natural systems. The zoo and bird sanctuary has initially been set up to entertain people. gradually, they have played an essential role in nature maintenance. The goal of the Zoo and Bird sanctuary is conservation. It was easy because of the research done to computerize the management, maintenance and ticketing of the wildlife. Therefore, this system called the management system “KALARAVA” was developed. This is a python based technology to manage birds.

### 1.OBJECTIVE OF THE PROJECT:

This application software can get information about Birds. The people don’t get aware about strange birds so they can get complete information with an ID. The Bird sanctuary System is a python based technology that manages people, birds, and details and provides tickets to those who visit the Bird sanctuary with their families.

This System adequately manages and processes all the works of the Bird sanctuary. The software system can store bird sanctuary visitor ticket data, reducing response time to queries from different users. This is A project based on the python language with a Django framework that manages people. We will provide tickets to those who go to the bird sanctuary with their families.

### 2.INPUT OF THE PROJECT:

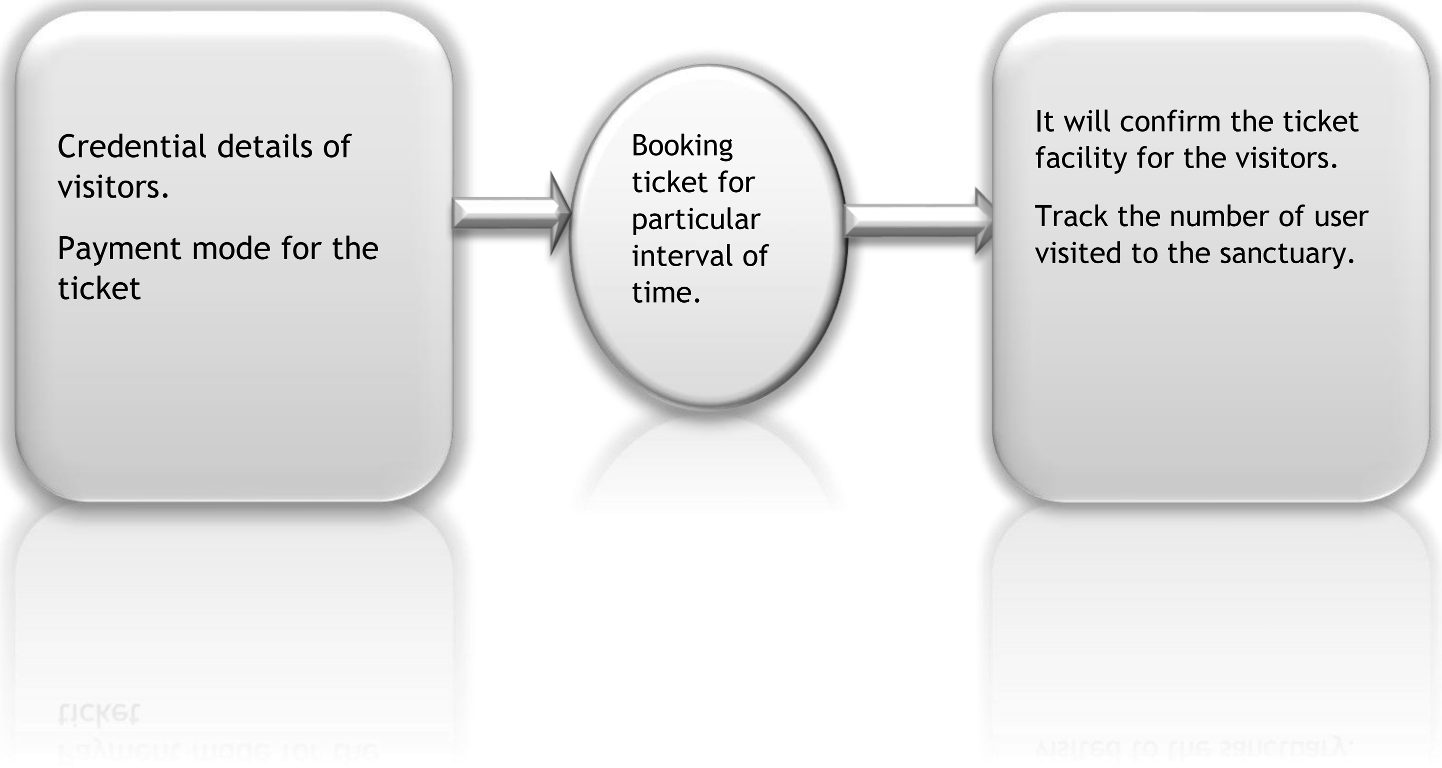
- The main input to the project is that the user login to the website “KALARAVA” by entering the credential details and then the system helps to book ticket to visit the Bird sanctuary.
- Allows administrators to view and edit regular adult and child ticket details.

## 3. OUTPUT OF THE PROJECT:

- This application provides a system to provide information about the visitors.
- It also manages Birds.
- The System also manages and calculates ticket prices to minimize the waiting time of the visitors .
- It can also be used to search for ticket IDs.

## 4.Process logic:

Input Process Output



Credential details of visitors.

Payment mode for the ticket

Booking ticket for particular interval of time.

It will confirm the ticket facility for the visitors.

Track the number of user visited to the sanctuary.

## System Requirement Specification

### Hardware Requirements

| SL.NO | HARDWARE | DESCRIPTICON |
|---|---|---|
| 1 | PROCESSOR | 11th Gen Intel®Core™ |
| 2 | RAM | 8.00GB and above |
| 3 | HARD DISK | 250GB of available hard disk space |
| 4 | PROCESSOR SPEED | 3.00GHz |

### Software Requirements

| SL.NO. | SOFTWARE | DESCRIPTION |
|---|---|---|
| 1 | SERVER-SIDE SCRIPT | Python |
| 2 | OPERATING SYSTEM | Windows 11 |
| 3 | DATABASE | XAMP, Django framework |
| 4 | FRONT END | HTML, CSS, JavaScript, Bootstrap,<br>PyCharm |
| 5 | WEB SERVER | XAMP |

# ANALYSIS

## a) System requirements:

System specification forms the foundation on which the architecture, design, and implementation of a software is built. Documents containing system specifications are critical because major requirements as a result of not having a specification document on hand. System specification documents can thus be defined as the requirements documentation that formally specifies the system-level requirements of a software application.

System specification documents most predominantly contain information on basic website requirements which include:

- performance levels
- reliability
- quality
- interfaces
- security and privacy
- constraints and limitations
- functional capabilities
- data structures and elements

### b) Existing System:

- In the present system all the works are non-computerized.
- It consumes more time and also needs more human efforts.
- In the existing system the people can get the ticket in offline mode only ,and they don't Have to clear there querries they don't know Particular timings of the birdsanctuary

- **c) Proposed Systems:**
- The proposed system is a web-based application that reduces the paper work.
- The proposed system influences or interacts with visitors to the Bird sanctuary.
- Allows administrators to view and edit regular adult and child ticket details.
- It also produces reports of people visiting the Bird sanctuary between appointments.
- It can also be used to search for ticket IDs.
- Display special events information, visitors' information and description
- Integration of records for all appointments.

## Modules:

- **ADMIN:**

- **Login: The admin can login to the system using a username and password.**
- **View Bird sanctuary:**
- **Accept/Reject: They can view the bird sanctuary details. Also, they can accept or reject the ticket allocation for the user.**
- **View User Information: The admin can view all the user information based on ticket allotment**

❖ USER:

- Register: user will need to register first to log in with their basic details.
- Login: They can log in using their username and password.
- My profile: Also, they can make necessary changes to their profile details.
- Change Password: They can easily change their old password to the new one.
- Book Ticket: They can further book the ticket for their families.
- Payment: The user can pay the fees through the online mode.

❖ Staff manager:

- Login: manager can login with credential details.
- View user information: the manager can view the number of visitors per day  visited the bird sanctuary and track the user based on the ticket allotment of the user.

## Limitations:

- Visitors will need to upload the data correctly, otherwise, it will be led to the wrong output.
- To keep the database up to date regular update is needed.
- Internet connection is mandatory.
- No multiple language support.

## Advantages:

- These sanctuaries help in the conservation of various species of birds.
- They help in the maintenance of the ecological balance.
- They help in the protection of the environment.
- They help in the propagation of various species of birds.
- They provide shelter to the birds.

## Future Enhancement of The Project

- Management of Birds and species.
- In future Android application can be developed.
- Using some more tools on AI, based on Bird chirping Bird can be recognized easily.

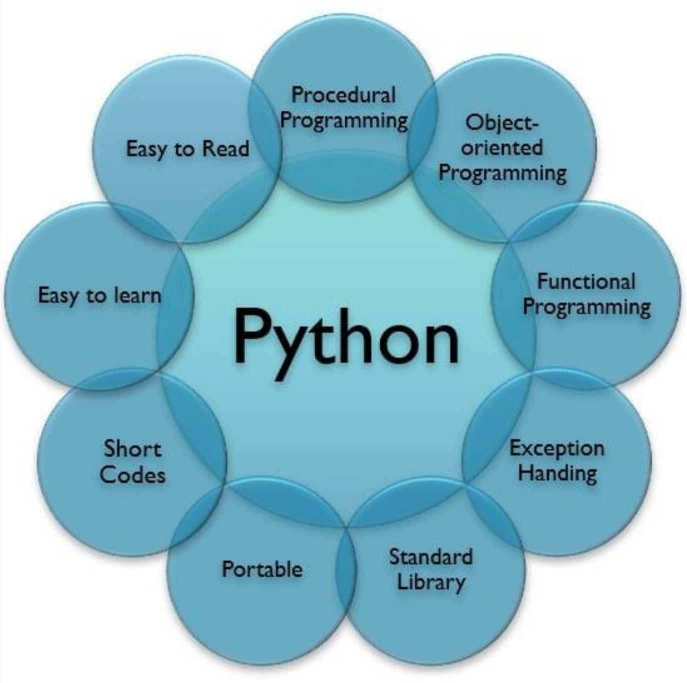
# FRAMEWORK

## A. PYTHON:

**Python** is a very popular general-purpose interpreted, interactive, object-oriented, and high-level programming language. Python is dynamically-typed and garbage-collected programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL).

Python supports multiple programming paradigms, including Procedural, Object Oriented and Functional programming language. Python design philosophy emphasizes code readability with the use of significant indentation.

### ADVANTAGES OF PYTHON:



- **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.

- **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
- **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** − Python provides interfaces to all major commercial databases.
- **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** − Python provides a better structure and support for large programs than shell scripting.

## B. DJANGO:

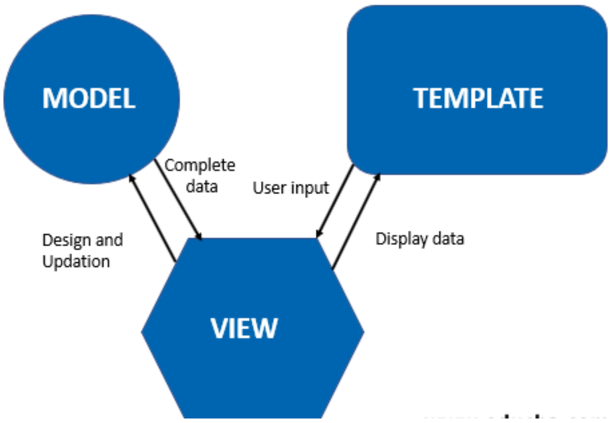
Django is a web application framework written in Python programming language. It is based on MVT (Model View Template) design pattern. The Django is very demanding due to its rapid development feature. It takes less time to build application after collecting client requirement.

This framework uses a famous tag line: **The web framework for perfectionists with deadlines.**

By using Django, we can build web applications in very less time. Django is designed in such a manner that it handles much of configure things automatically, so we can focus on application development only.

### Architecture of Django:

Django Architecture follows the MVT structure. In MVT, M stands for Model, V stands for views, and T stands for Templates. Model is the structure of storing the data in the database, the view is a python function used to handle the web request,



## C.HTML:

HTML stands for Hyper Text Markup Language, which is the most widely used language on Web to develop web pages. HTML was created by Berners-Lee in late 1991 but "HTML 2.0" was the first standard HTML specification which was published in 1995. HTML 4.01 was a major version of HTML and it was published in late 1999.

**Syntax:**

```
<! DOCTYPE html>

<html>
        <head>
        <title>This is a Title </title>
        </head>
        <body>
<p>Hello World! </p>
        </body>
</html>
```

**<! DOCTYPE>**

This tag defines the document type and HTML version.

**<HTML>**

This tag encloses the complete HTML document and mainly comprises of document header which is represented by <head>...</head> and document body which is represented by <body>...</body> tags.

**<HEAD>**

This tag represents the document's header which can keep other HTML tags like <title>, <link> etc.

**<TITLE>**

The <title> tag is used inside the <head> tag to mention the document title.

**<BODY>**This tag represents the document's body which keeps other HTML tags like<h1>, <div>, <p> etc.

## D. Cascading Style Sheet (CSS):

**Cascading Style Sheets** was invented by **HakonWium Lie** on October 7, 1994 and maintained through a group of people within the W3C called the CSS Working Group

CSS is the language for describing the presentation of Web pages, including colors, layout, and fonts. It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. CSS is independent of HTML and can be used with any XML-based markup language. The separation of HTML from CSS makes it easier to maintain sites, share style sheets across pages, and tailor pages to different environments. There are three types CSS mainly

- External CSS
- Internal CSS
- Inline CSS

## E. JAVASCRIPT:

**JavaScript** is a dynamic computer programming language. It is most commonly used as part of Web browsers, whose implementations allow client-side **scripts** to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed.

## F. What is XAMPP?

- XAMPP is an abbreviation where *X stands for Cross-Platform, A stands for Apache, M stands for* ***MYSQL****, and the Ps stand for PHP and Perl*, respectively. It is an open-source package of web solutions that includes Apache distribution for many servers and command-line executables along with modules such as Apache server, MariaDB, PHP, and Perl.
- XAMPP helps a local host or server to test its website and clients via computers and laptops before releasing it to the main server. It is a platform that furnishes a suitable environment to test and verify the working of projects based on Apache, Perl, MySQL database, and PHP through the system of the host itself. Among these technologies, Perl is a programming language used for web development, PHP is a backend scripting language, and MariaDB is the most vividly used database developed by MySQL.

## G. BOOTSTRAP:

Bootstrap was developed by Mark Otto and Jacob Thornton at Twitter. It was released as an open-source product in August 2011 on GitHub.Bootstrap is the most popular HTML, CSS and JavaScript framework for developing a responsive and mobile friendly website. It is absolutely free to download and use. It is a front-end framework used for easier and faster web development. It includes HTML and CSS based design templates for typography, forms, buttons, tables, navigation, modals, image carousels and many others.

## Python IDE:

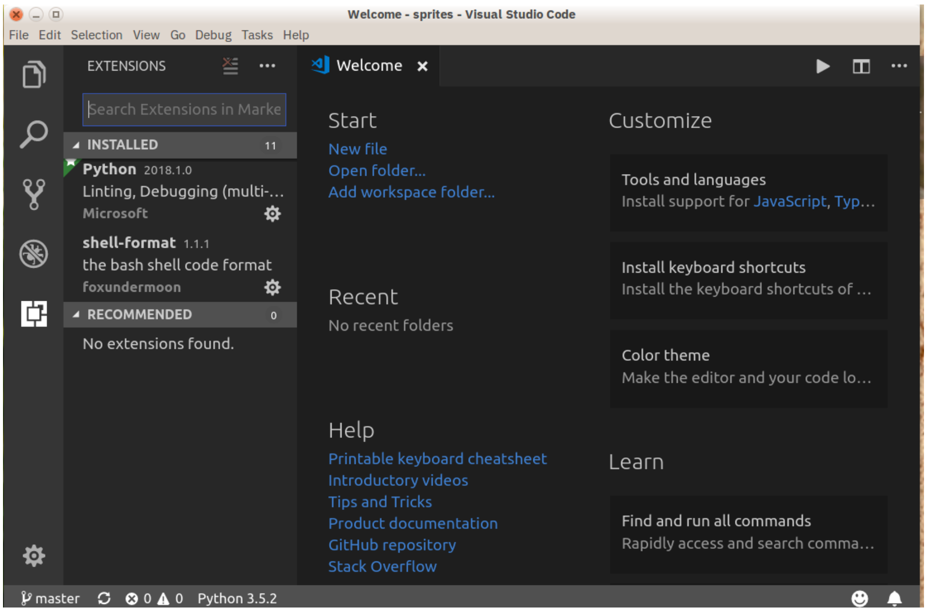
An IDE (or Integrated Development Environment) is a program dedicated to software development. As the name implies, IDEs integrate several tools specifically designed for software development. These tools usually include:

- An editor designed to handle code (with, for example, syntax highlighting and auto-completion)
- Build, execution, and debugging tools
- Some form of source control

### Visual Studio Code:

**Category:** Code Editor
**Website:** https://code.visualstudio.com/
**Python tools:** https://marketplace.visualstudio.com/items?itemName=ms-python.python

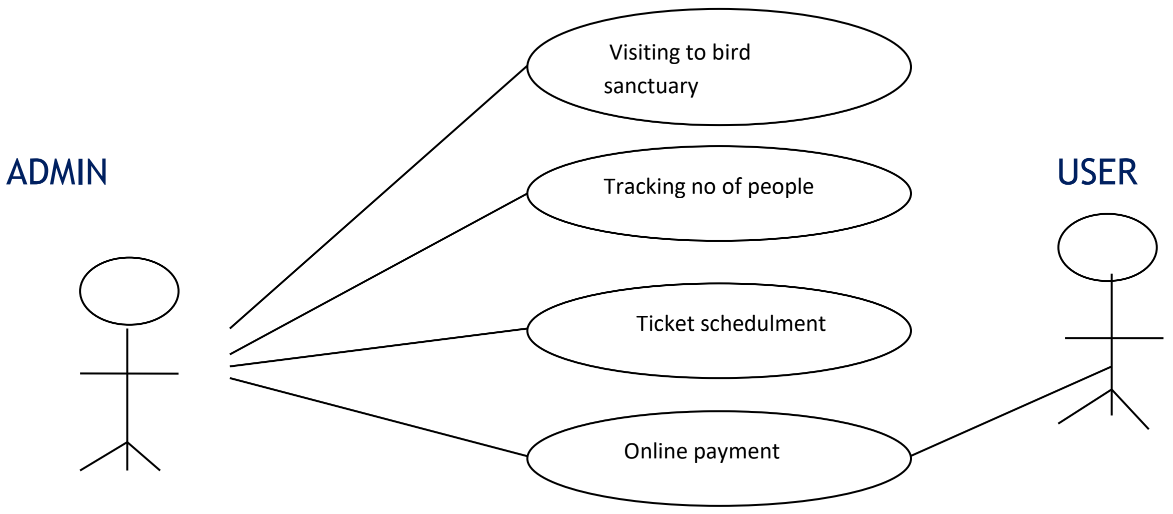
Installing Python support in VS Code is very accessible: the Marketplace is a quick button click away. Search for Python, click Install, and restart if necessary. VS Code will recognize your Python installation and libraries automatically.


Welcome - sprites - Visual Studio Code
File Edit Selection View Go Debug Tasks Help
EXTENSIONS
Search Extensions in Marke
INSTALLED 11
Python 2018.1.0
Linting, Debugging (multi-...
Microsoft
shell-format 1.1.1
the bash shell code format
foxundermoon
RECOMMENDED 0
No extensions found.
Welcome
Start
New file
Open folder...
Add workspace folder...
Recent
No recent folders
Help
Printable keyboard cheatsheet
Introductory videos
Tips and Tricks
Product documentation
GitHub repository
Stack Overflow
Customize
Tools and languages
Install support for JavaScript, Typ...
Install keyboard shortcuts
Install the keyboard shortcuts of ...
Color theme
Make the editor and your code lo...
Learn
Find and run all commands
Rapidly access and search comma...
master
0 0
Python 3.5.2

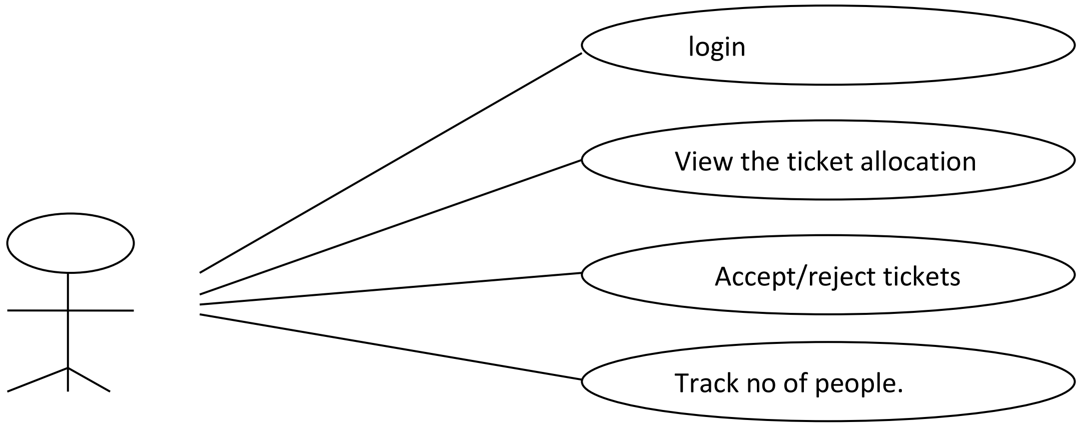
# Use Case Diagrams

**USE CASES DESCRIPTION:**

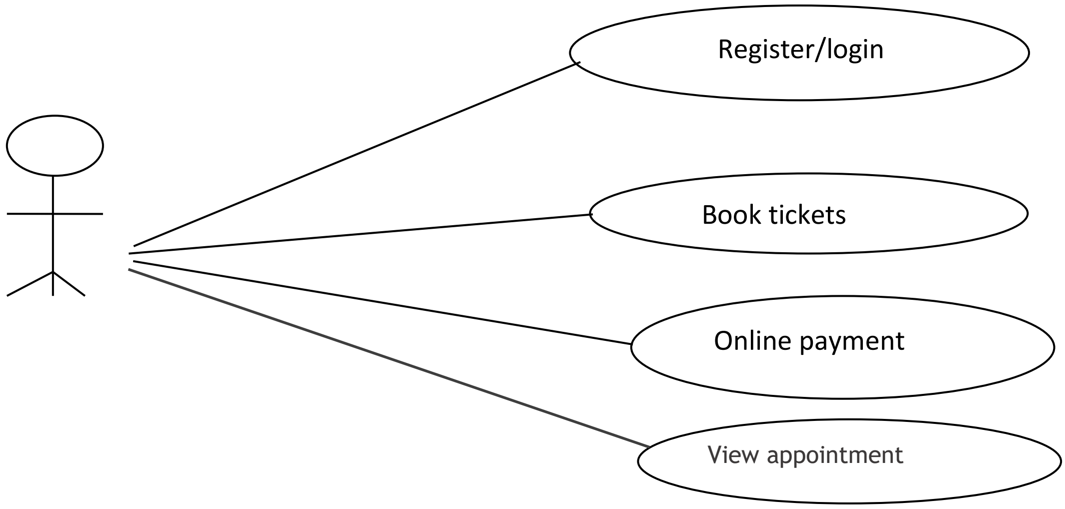
A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal. Use case is a list of steps, typically defining interactions between a role (known in UML as an actor) and a system, to achieve a goal. The actor can be a human or an external system.



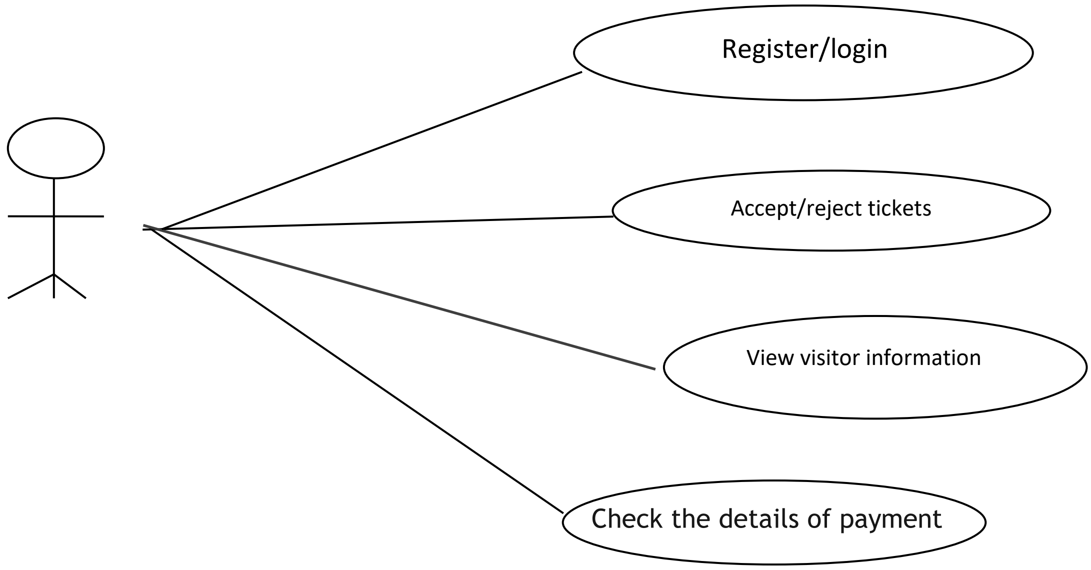
## ADMIN:


login
View the ticket allocation
Accept/reject tickets
Track no of people.

## USER:


Register/login
Book tickets
Online payment
View appointment

## STAFF MANAGER:



# Software Requirement Specification (SRS)

AIM:

PREPARE A SOFTWARE REQUIREMENT SPECIFICATION (SRS) DOCUMENT FOR A GIVEN PROBLEM STATEMENT.

PROBLEM STATEMENT:

THIS PROJECT IS TO DEVELOP A WEB APPLICATION FOR THE BIRDSANCTUARY. IT DESCRIBE THE FUNCTIONS AND TASKS THAT THE SYSTEM CAN PERFORM. THIS IS USED TO DESCRIBE THE SCOPE OF THE PROJECT AND TO PLAN FOR THE SYSTEM DESIGN AND IMPLEMENTATION. IT KEEPS DETAILS AND HISTORY OF A VISITORS AND THEIR DETAILS. IT GENERATES THE TICKETS FOR THE VISITORS TO VISIT THE BIRDSANCTUARY. THE SOFTWARE HANDLES MOST NECESSARY DETAILS.

OBJECTIVES:

TO UNDERSTAND DIFFERENT SECTIONS OF SOFTWARE REQUIREMENT SPECIFICATION (SRS).

- TO UNDERSTAND FUNCTIONAL REQUIREMENTS OF THE SYSTEM.
- TO UNDERSTAND PERFORMANCE REQUIREMENTS OF THE SYSTEM.
- TO APPLY DESIGN CONSTRAINTS AND APPROPRIATE VALIDATION ON THE WEB APPLICATION.

THEORY:

SOFTWARE REQUIREMENT SPECIFICATION (SRS) DOCUMENT USUALLY CONTAINS A BIRD SANCTUARY DETAILS BY UNDERSTANDING THE VISITOR'S INFORMATION. IT DESCRIBES THE NON-FUNCTIONS GOALS AND TASKS THAT THE SYSTEM CAN PERFORM.THIS IS USED TO DESCRIBE THE SCOPE OF THE PROJECT AND TO PLAN FOR THE SYSTEMS DESIGN AND IMPLENTATION.

## Introduction

### Purpose:

**The software is for automation of bird sanctuary details. The software includes maintaining visitors' details providing and maintaining the ticket for the visitors by accessing the payment.**

### Scope of development project:

The proposed software is the bird sanctuary system the system will be used to track the number of visitors visited to the bird sanctuary and then storing that data for future usage the current system using the online payment for the generation of ticket hence its saving lots of time to get the ticket for the visitors visits the bird sanctuary. requirement statement in this document is both functional and non-functional.

### References

- Books
- Websites

## Overall Description:

### Product Functions

The bird sanctuary projects help to track the number of visitors visited to the bird sanctuary. this website helps to maintain the visitor information and also online payment facility to get the ticket to visit the sanctuary.

The main purpose of this project is to reduce the manual work this web application is capable of **maintaining visitor information, online payment facility for the visitors to get the ticket.**

**Operating Environment:**

The product will be operating in windows environment. The basic input devices required are keyboard, mouse and output devices are monitor, printer etc.

**Development Environment:**

**Software Configuration:**

This web application uses,

Operating system: windows 11

Language: HTML, CSS, JavaScript, Bootstrap (Front end)

framework: Django

Server-Side Script: python

Web server: Xamp Server

**Hardware Configuration:**

Processor: 11th Gen Intel(R) Core (TM)

Processor Speed: 3.00GHz

RAM: 8.00GB

Hard Disk: 20GB free space

### Data Requirement

### Visitors credential details:

The inputs consist of the query to the database and output consists of the solutions for the query. In this project the inputs will be the queries as fired by the visitors' credential details. Keeping an accurate database of visitors and their ticket details.

### External Interface Requirements

The purpose of this section is to identify and document interfaces and interaction of the website with external entities in detail.

The website provides good graphical interface for the owners/managers who can operate on the system, performing the required task such as create, update, viewing the details of the ticket allotment for visitors. The website should provide payment option in online as well as offline mode.

### Functional requirements

Functional requirements are the following:

1. User/visitor will be able to add their profile and mode of payment and particular time and date to visit the sanctuary.
2. For each visitor, generating the ticket to visit the bird sanctuary.
3. The visitor will able to select the mode of the payment.
4. It will give an option to cancel the appointment of the booking.
5. There should be a facility of AI chatbot to communicate with the system easily.

**System Features:**

The Bird sanctuary should be providing the surety that their data is secure. This is possible by providing:

- visitor authentication and validation using their unique visitor id.
- Proper monitoring by the admin which includes updating data.
- Proper ticket booking details are maintained for every visitor.

**Other Non-functional Requirements**

**Performance Requirement**

The proposed website that we are going to develop will be used as the Chief performance system by particular bird sanctuary which interacts with visitors regarding their ticket booking and payment.

Therefore, it is expected that the bird sanctuary would perform functionally all the requirements that are required by the visitors.

- The performance of the system should be fast accurate.

This system must support many people at a time the user interface screen shall response within time. The system must confirm to the Microsoft accessibility.

**Safety Requirements**

- Website will use secured database.
- Normal users can just read information but they cannot edit or modify except their personal and some other information.
- System will have different types of users and every user has access constraints.
- Proper user authentication should be provided.

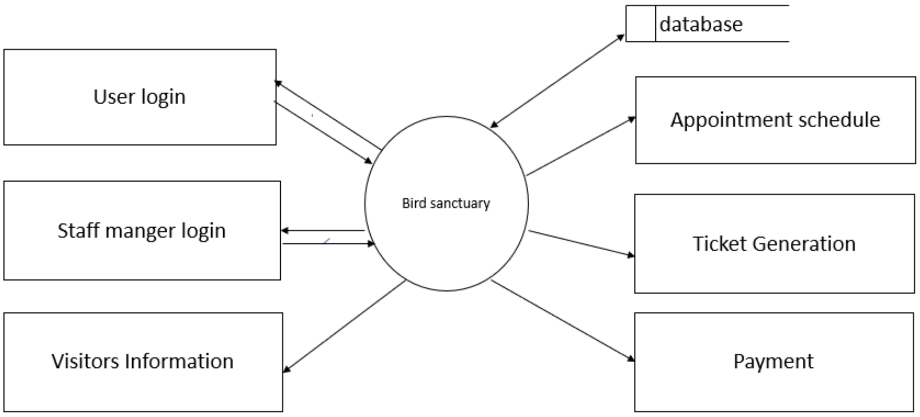
## Conclusion:

This SRS document is used to give details regarding bird sanctuary details in this all functional and non- functional requirement are specified in order to get clear cut idea to develop a project.

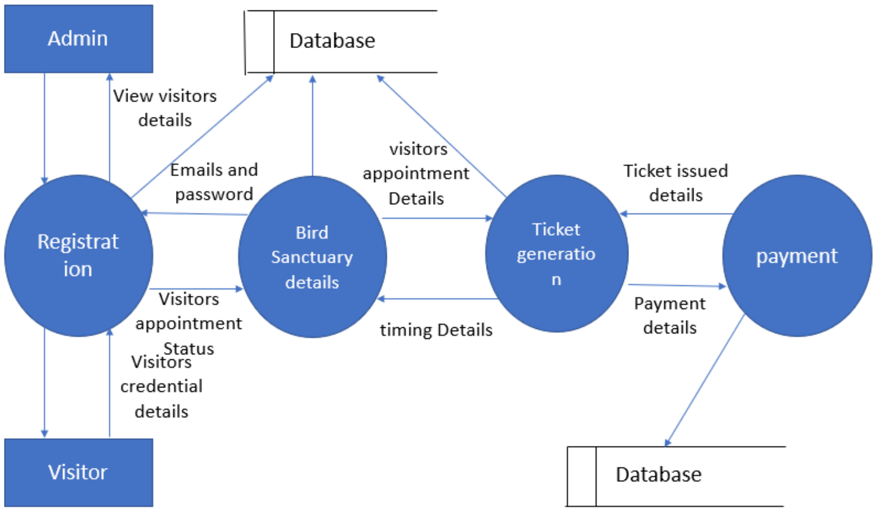
# DATA FLOW DIAGRAM

A data flow diagram (DFD) is a significant modeling technique for analyzing and constructing information processes. Data-flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. DFDs can also be used for the visualization of data processing (structured design). On a DFD, data items flow from an external data source or an internal data store to an Internal data store or an external data sink, via an internal process. A DFD provides no information about the timing or ordering of processes, or about whether processes will operate in sequence or in parallel. It is therefore quite different from a flowchart.

**Level Zero DFD:**



## Level One DFD:



# Entity Relationship Diagram (ER DIAGRAM)

An entity-relationship diagram is a data modelling technique that creates a graphical representation of the entities, and the relationships between entities, within an information system. An entity-relationship model (ERM) is an abstract and conceptual representation of data. Entity-relationship modelling is a database modelling method, used to produce a type of conceptual schema or semantic data model of a system, often a relational database, and its requirements in a top-down fashion. Diagrams created by this process are called entity-relationship diagrams, ER diagrams, or ERDs.

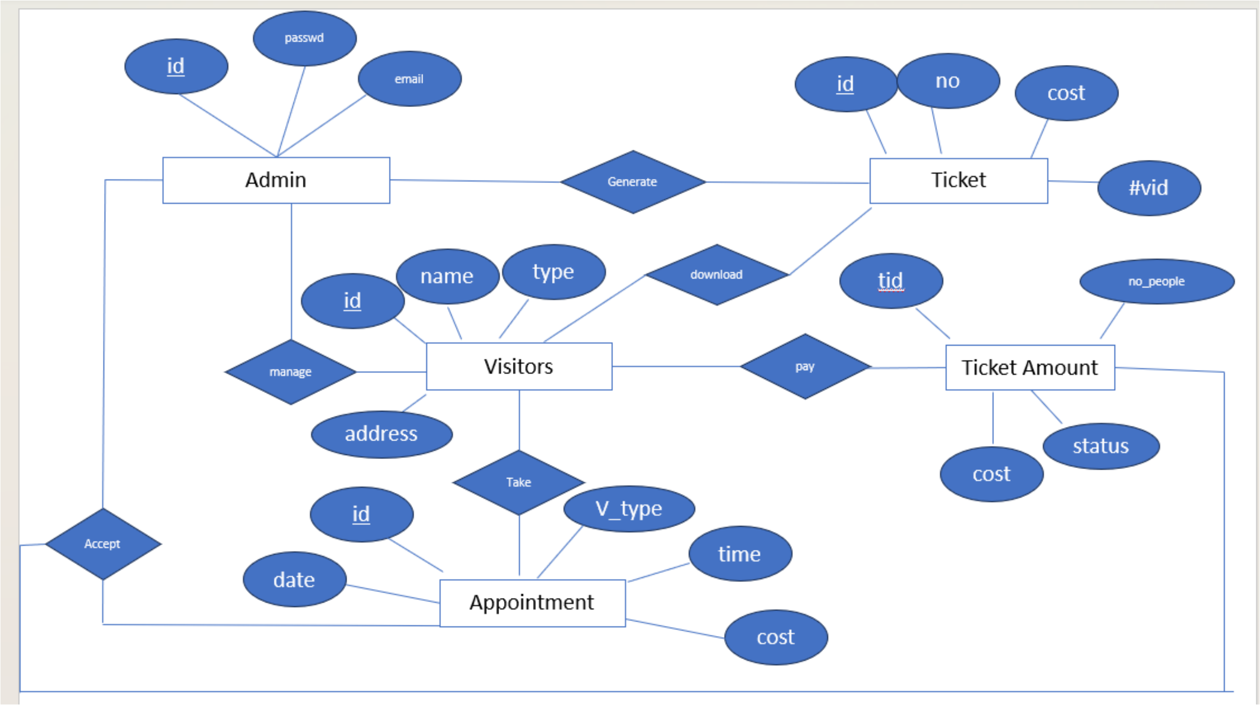
**The three main components of an E-R Diagram are:**

- The entity is a person, object, place or event for which data is collected. For example, if you consider the information system for a business, entities would include not only customers, but the customer's address, and orders as well. The entity is represented by a rectangle and labelled with a singular noun.
- The relationship is the interaction between the entities. In the example above, the customer places an order, so the word "places" defines the relationship between that instance of a customer and the order or orders that they place. A relationship may be represented by a diamond

shape, or more simply, by the line connecting the entities. In either case, verbs are used to label the relationships.

- The cardinality defines the relationship between the entities in terms of numbers. An entity may be optional: for example, a sales representative could have no customers or could have one or many customers; or mandatory: for example, there must be at least one product listed in an order. There are several different types of cardinality notations; crow's foot notation, used here, is a common one. In crow's foot notation, a single bar indicates one, a double bar indicates one and only one (for example, a single instance of a product can only be stored in one warehouse), a circle indicates zero, and a crow's foot indicates many. The three main cardinal relationships are: one-to-one, expressed as 1:1; one-to-many, expressed as 1: N; and many-to-many, expressed as M: N.

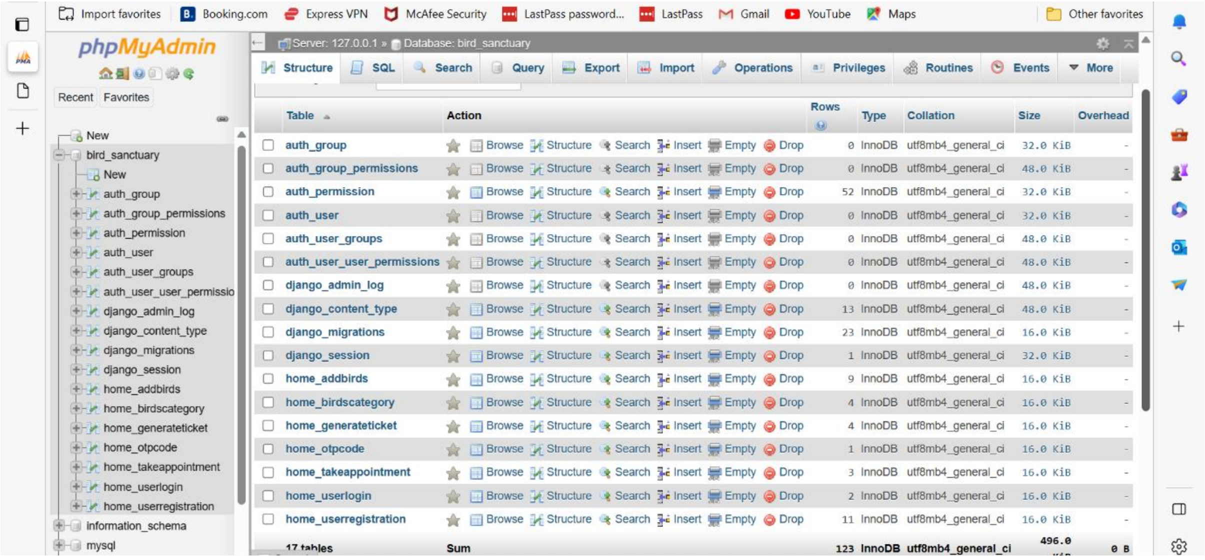
## E-R Diagram for Bird Sanctuary System:



# SYSTEM DEVELOPMENT

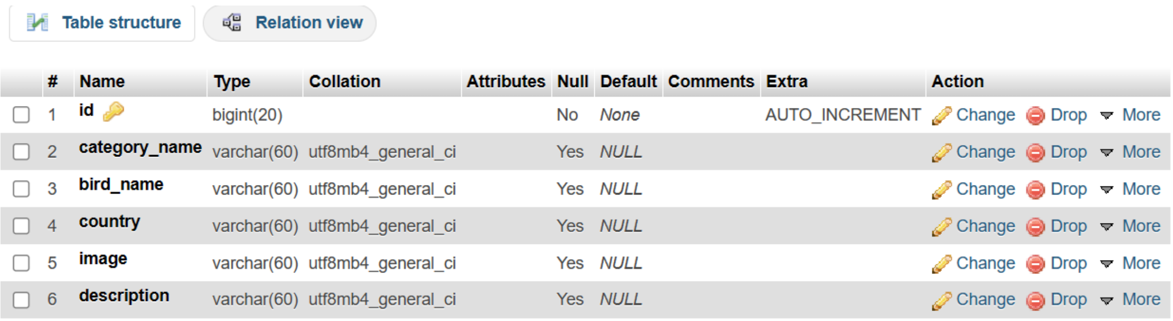
The development phase focuses on how, that is, during development a software engineer attempts to define how data are to be structured, how function is to be implemented within a software architecture, how procedural details are to be implemented, how interfaces are to be characterized and how testing will be performed.

## TABLES USED IN DATABASE:



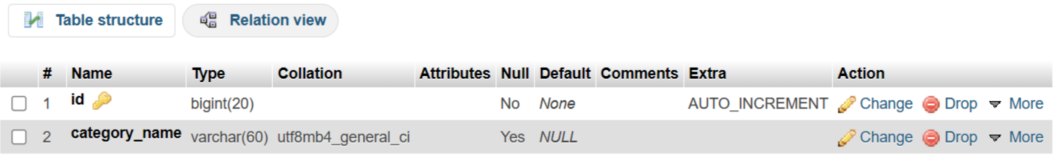
| Table | Action | Rows | Type | Collation | Size | Overhead |
|---|---|---|---|---|---|---|
| auth_group | ☆ Browse Structure Search Insert Empty Drop | 0 | InnoDB | utf8mb4_general_ci | 32.0 KiB | - |
| auth_group_permissions | ☆ Browse Structure Search Insert Empty Drop | 0 | InnoDB | utf8mb4_general_ci | 48.0 KiB | - |
| auth_permission | ☆ Browse Structure Search Insert Empty Drop | 52 | InnoDB | utf8mb4_general_ci | 32.0 KiB | - |
| auth_user | ☆ Browse Structure Search Insert Empty Drop | 0 | InnoDB | utf8mb4_general_ci | 32.0 KiB | - |
| auth_user_groups | ☆ Browse Structure Search Insert Empty Drop | 0 | InnoDB | utf8mb4_general_ci | 48.0 KiB | - |
| auth_user_user_permissions | ☆ Browse Structure Search Insert Empty Drop | 0 | InnoDB | utf8mb4_general_ci | 48.0 KiB | - |
| django_admin_log | ☆ Browse Structure Search Insert Empty Drop | 0 | InnoDB | utf8mb4_general_ci | 48.0 KiB | - |
| django_content_type | ☆ Browse Structure Search Insert Empty Drop | 13 | InnoDB | utf8mb4_general_ci | 48.0 KiB | - |
| django_migrations | ☆ Browse Structure Search Insert Empty Drop | 23 | InnoDB | utf8mb4_general_ci | 16.0 KiB | - |
| django_session | ☆ Browse Structure Search Insert Empty Drop | 1 | InnoDB | utf8mb4_general_ci | 32.0 KiB | - |
| home_addbirds | ☆ Browse Structure Search Insert Empty Drop | 9 | InnoDB | utf8mb4_general_ci | 16.0 KiB | - |
| home_birdscategory | ☆ Browse Structure Search Insert Empty Drop | 4 | InnoDB | utf8mb4_general_ci | 16.0 KiB | - |
| home_generateticket | ☆ Browse Structure Search Insert Empty Drop | 4 | InnoDB | utf8mb4_general_ci | 16.0 KiB | - |
| home_otpcode | ☆ Browse Structure Search Insert Empty Drop | 1 | InnoDB | utf8mb4_general_ci | 16.0 KiB | - |
| home_takeappointment | ☆ Browse Structure Search Insert Empty Drop | 3 | InnoDB | utf8mb4_general_ci | 16.0 KiB | - |
| home_userlogin | ☆ Browse Structure Search Insert Empty Drop | 2 | InnoDB | utf8mb4_general_ci | 16.0 KiB | - |
| home_userregistration | ☆ Browse Structure Search Insert Empty Drop | 11 | InnoDB | utf8mb4_general_ci | 16.0 KiB | - |
| 17 tables | Sum | 123 | InnoDB | utf8mb4_general_ci | 496.0 | 0 B |

## 1.add birds table



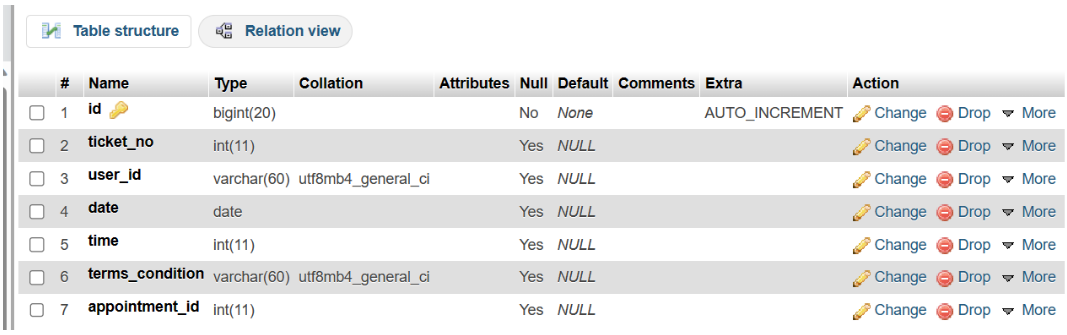
| | # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|---|
| □ | 1 | **id** | bigint(20) | | | No | *None* | | AUTO_INCREMENT | Change Drop More |
| □ | 2 | **category_name** | varchar(60) | utf8mb4_general_ci | | Yes | *NULL* | | | Change Drop More |
| □ | 3 | **bird_name** | varchar(60) | utf8mb4_general_ci | | Yes | *NULL* | | | Change Drop More |
| □ | 4 | **country** | varchar(60) | utf8mb4_general_ci | | Yes | *NULL* | | | Change Drop More |
| □ | 5 | **image** | varchar(60) | utf8mb4_general_ci | | Yes | *NULL* | | | Change Drop More |
| □ | 6 | **description** | varchar(60) | utf8mb4_general_ci | | Yes | *NULL* | | | Change Drop More |

## 2.Birdscategory Table



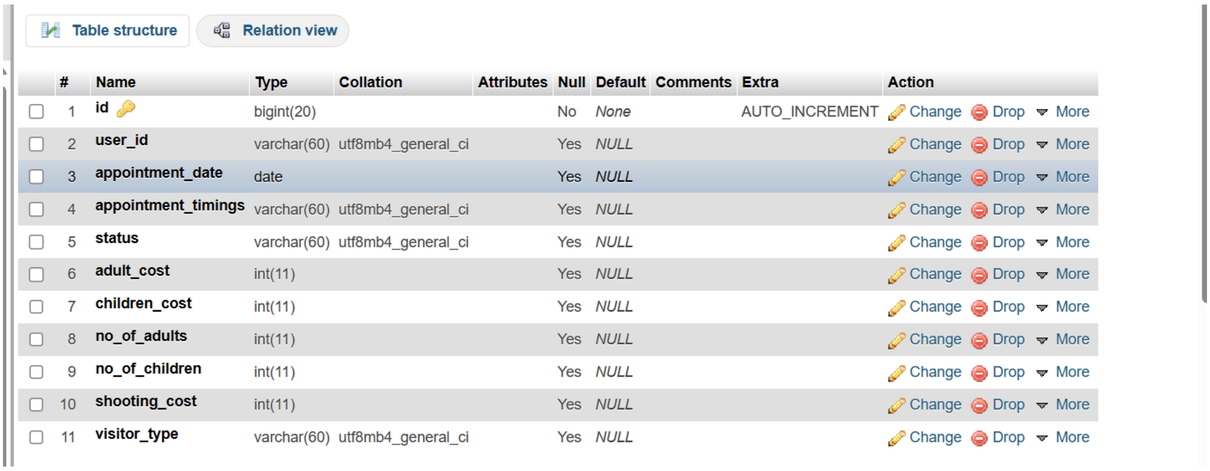
| | # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|---|
| ☐ | 1 | **id** | bigint(20) | | | No | *None* | | AUTO_INCREMENT | Change Drop More |
| ☐ | 2 | **category_name** | varchar(60) | utf8mb4_general_ci | | Yes | *NULL* | | | Change Drop More |

## 3.GenerateTicket Table



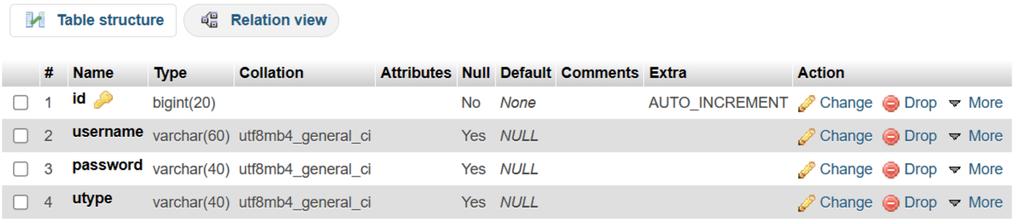
| | # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|---|
| ☐ | 1 | **id** | bigint(20) | | | No | *None* | | AUTO_INCREMENT | Change Drop More |
| ☐ | 2 | **ticket_no** | int(11) | | | Yes | *NULL* | | | Change Drop More |
| ☐ | 3 | **user_id** | varchar(60) | utf8mb4_general_ci | | Yes | *NULL* | | | Change Drop More |
| ☐ | 4 | **date** | date | | | Yes | *NULL* | | | Change Drop More |
| ☐ | 5 | **time** | int(11) | | | Yes | *NULL* | | | Change Drop More |
| ☐ | 6 | **terms_condition** | varchar(60) | utf8mb4_general_ci | | Yes | *NULL* | | | Change Drop More |
| ☐ | 7 | **appointment_id** | int(11) | | | Yes | *NULL* | | | Change Drop More |

## 4.Takeappointmate Table



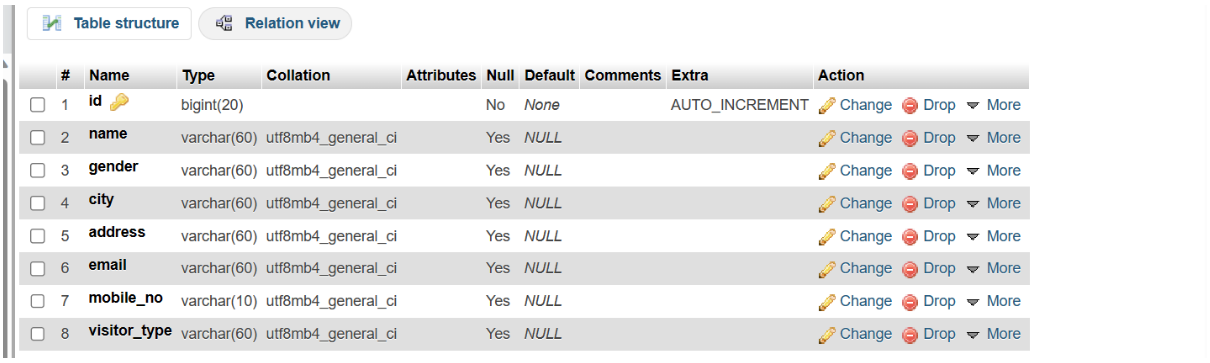
| | # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|---|
| ☐ | 1 | **id** | bigint(20) | | | No | *None* | | AUTO_INCREMENT | Change Drop More |
| ☐ | 2 | **user_id** | varchar(60) | utf8mb4_general_ci | | Yes | *NULL* | | | Change Drop More |
| ☐ | 3 | **appointment_date** | date | | | Yes | *NULL* | | | Change Drop More |
| ☐ | 4 | **appointment_timings** | varchar(60) | utf8mb4_general_ci | | Yes | *NULL* | | | Change Drop More |
| ☐ | 5 | **status** | varchar(60) | utf8mb4_general_ci | | Yes | *NULL* | | | Change Drop More |
| ☐ | 6 | **adult_cost** | int(11) | | | Yes | *NULL* | | | Change Drop More |
| ☐ | 7 | **children_cost** | int(11) | | | Yes | *NULL* | | | Change Drop More |
| ☐ | 8 | **no_of_adults** | int(11) | | | Yes | *NULL* | | | Change Drop More |
| ☐ | 9 | **no_of_children** | int(11) | | | Yes | *NULL* | | | Change Drop More |
| ☐ | 10 | **shooting_cost** | int(11) | | | Yes | *NULL* | | | Change Drop More |
| ☐ | 11 | **visitor_type** | varchar(60) | utf8mb4_general_ci | | Yes | *NULL* | | | Change Drop More |

## 5.User Login Table



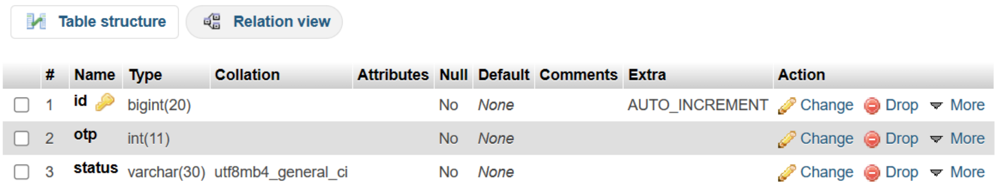
| | # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|---|
| ☐ | 1 | **id** | bigint(20) | | | No | *None* | | AUTO_INCREMENT | Change Drop More |
| ☐ | 2 | **username** | varchar(60) | utf8mb4_general_ci | | Yes | *NULL* | | | Change Drop More |
| ☐ | 3 | **password** | varchar(40) | utf8mb4_general_ci | | Yes | *NULL* | | | Change Drop More |
| ☐ | 4 | **utype** | varchar(40) | utf8mb4_general_ci | | Yes | *NULL* | | | Change Drop More |

## 6.User Registration Table



| | # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|---|
| □ | 1 | **id** 🔑 | bigint(20) | | | No | *None* | | AUTO_INCREMENT | Change Drop More |
| □ | 2 | **name** | varchar(60) | utf8mb4_general_ci | | Yes | *NULL* | | | Change Drop More |
| □ | 3 | **gender** | varchar(60) | utf8mb4_general_ci | | Yes | *NULL* | | | Change Drop More |
| □ | 4 | **city** | varchar(60) | utf8mb4_general_ci | | Yes | *NULL* | | | Change Drop More |
| □ | 5 | **address** | varchar(60) | utf8mb4_general_ci | | Yes | *NULL* | | | Change Drop More |
| □ | 6 | **email** | varchar(60) | utf8mb4_general_ci | | Yes | *NULL* | | | Change Drop More |
| □ | 7 | **mobile_no** | varchar(10) | utf8mb4_general_ci | | Yes | *NULL* | | | Change Drop More |
| □ | 8 | **visitor_type** | varchar(60) | utf8mb4_general_ci | | Yes | *NULL* | | | Change Drop More |

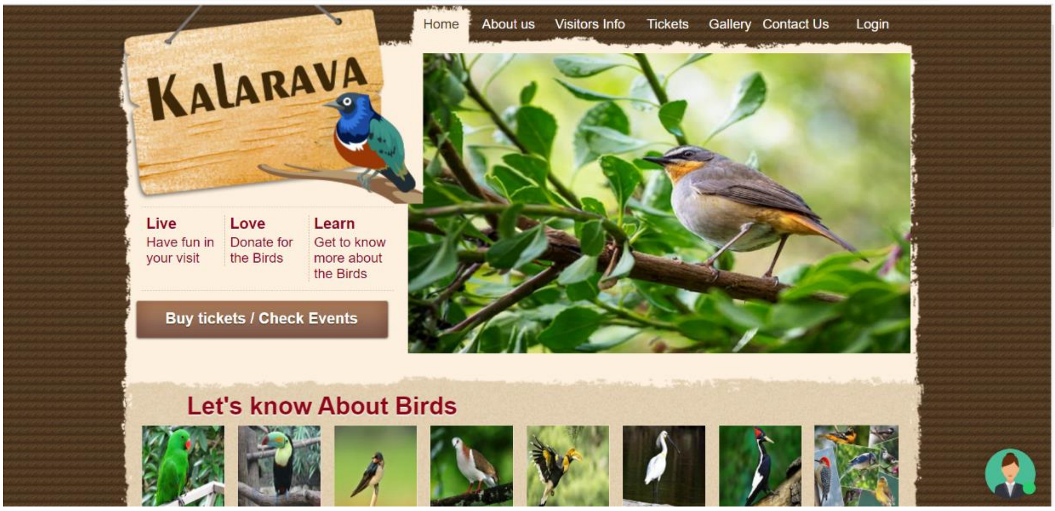
## 7.OTP Table



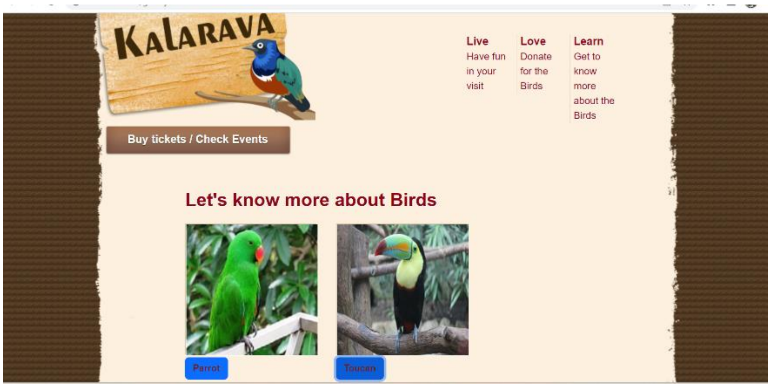
| | # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|---|---|---|---|---|---|---|---|---|---|
| □ | 1 | **id** 🔑 | bigint(20) | | | No | *None* | | AUTO_INCREMENT | Change Drop More |
| □ | 2 | **otp** | int(11) | | | No | *None* | | | Change Drop More |
| □ | 3 | **status** | varchar(30) | utf8mb4_general_ci | | No | *None* | | | Change Drop More |

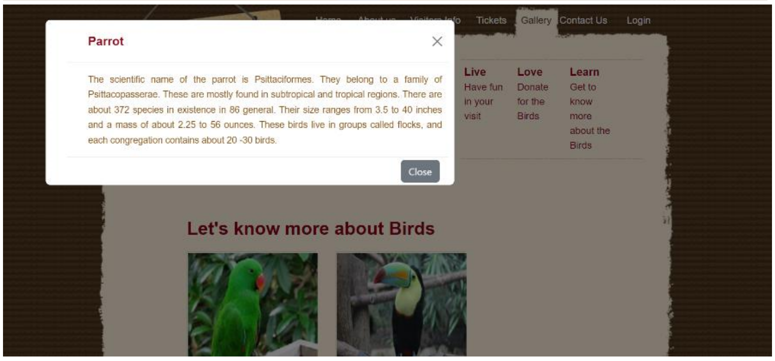
# SCREENSHOTS AND SOURCECODE

## HOME PAGE:



## View of birds information page:


KALARAVA
Buy tickets / Check Events
Live
Have fun in your visit
Love
Donate for the Birds
Learn
Get to know more about the Birds
Let's know more about Birds
Parrot
Toucan

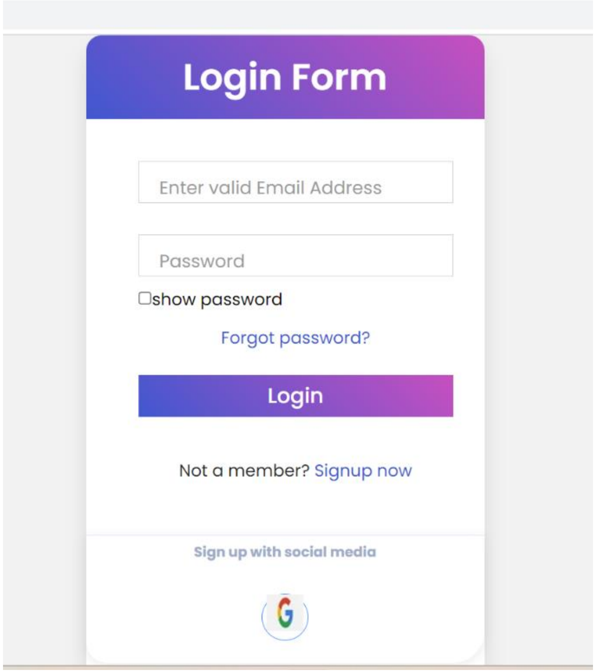


Parrot

The scientific name of the parrot is Psittaciformes. They belong to a family of Psittacopasserae. These are mostly found in subtropical and tropical regions. There are about 372 species in existence in 86 general. Their size ranges from 3.5 to 40 inches and a mass of about 2.25 to 56 ounces. These birds live in groups called flocks, and each congregation contains about 20 -30 birds.

Close

# Admin and User login page:



Login Form

Enter valid Email Address

Password

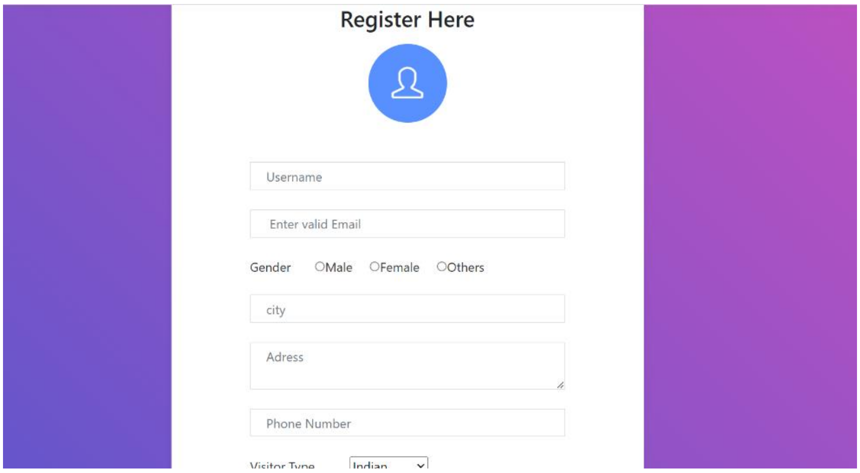
□show password

Forgot password?

Login

Not a member? Signup now

Sign up with social media



Register Here

Username

Enter valid Email

Gender ○Male ○Female ○Others

city

Adress

Phone Number

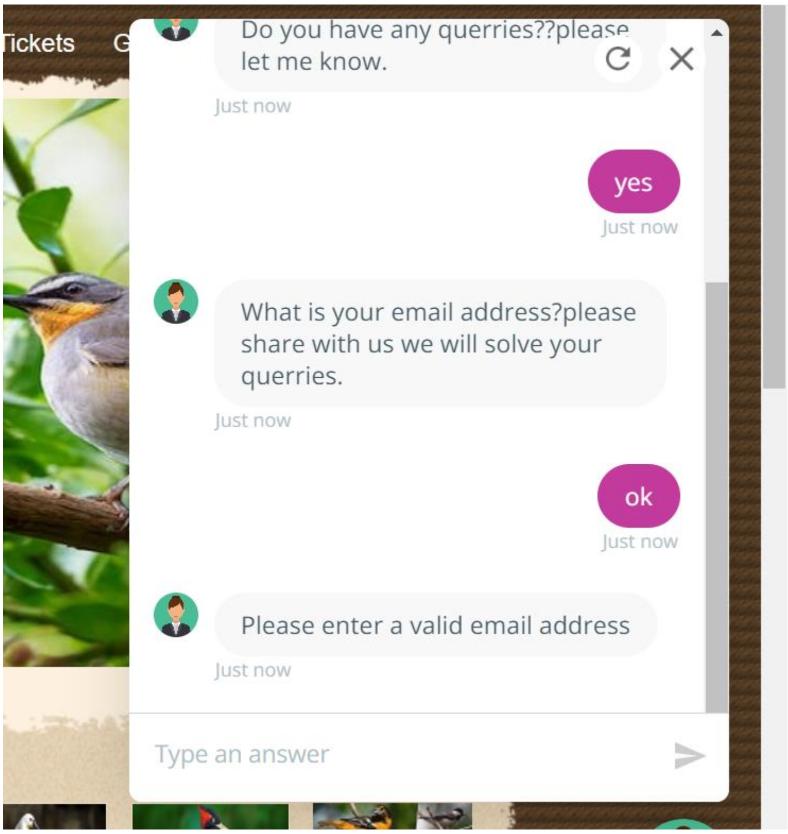
Visitor Type Indian

```
<div class="wrapper">
 <div class="title">
   Login Form
    </div>
   <form method="post">
     {% csrf_token %}
    <p style="color:red"> {{msg}}</p>
     <div class="field">
      <input type="email" name="t1" required>
      <label> Enter valid Email Address</label>
      </div>
       <div class="field">
       <input type="password" id="myInput1" name="t2" required>
```

```
<label>Password</label>
   </div>
   <div >
   <input type="checkbox" onclick="myFunction()"/>show password
     </div>
      <div class="content">
       <div class="pass-link">
      <a href="{% url 'forgotpass' %}">Forgot password?</a>
       </div>
        </div>
        <div class="field">
        <input type="submit" value="Login">
         </div>
            <div class="signup-link">
               Not a member? <a href="{% url 'reg'%}">Signup now</a>
            </div>
            </form>
       <div class="social-media">
        <h5 align="center">Sign up with social media</h5>
         <div class="social-icons">
         <a href="#"><i class="icon-social-google" title="Google">
         <img src="{% static '/images/google-img.jpg'%}"></i></a>
         </div></div>
          </body>
  <script>
    function myFunction()
    {
      var x = document. getElementById("myInput1");
      var y = document.getElementById("myInput2");
      var z = document. getElementById("myInput3");
      var n = document. getElementById("myInput4");

      if (x. type==="password")
      {
        x.type="text";
      }
      else
       {
        x.type = "password";
      }
    }
  </script>
</html>
```

## AI chatbot Assistance:



Do you have any querries??please let me know.

Just now

yes

Just now

What is your email address?please share with us we will solve your querries.

Just now

ok

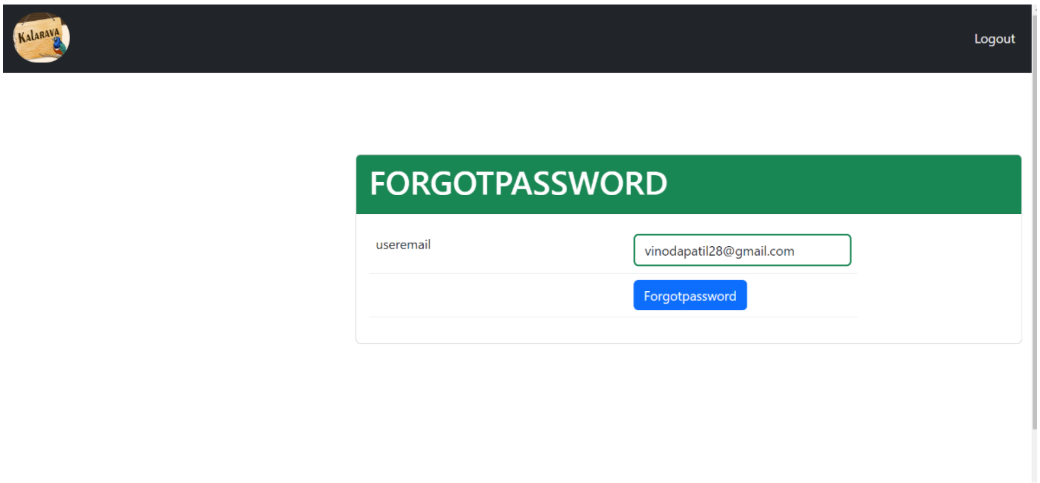
Just now

Please enter a valid email address

Just now

```
<script src="{% static 'js/script.js'%}" type="text/javascript">
 function (w, d) {w. CollectId = "645e611d1b0491e28f6193f2";
var h = d.head || d.getElementsByTagName("head")[0];
var s = d.createElement("script"); s.setAttribute("type", "text/javascript");
s.async=true; s.setAttribute("src",
" https://collectcdn.com/launcher.js"); h.appendChild(s); })(window,
document);
</script>
```

## Forgot Password:



FORGOTPASSWORD

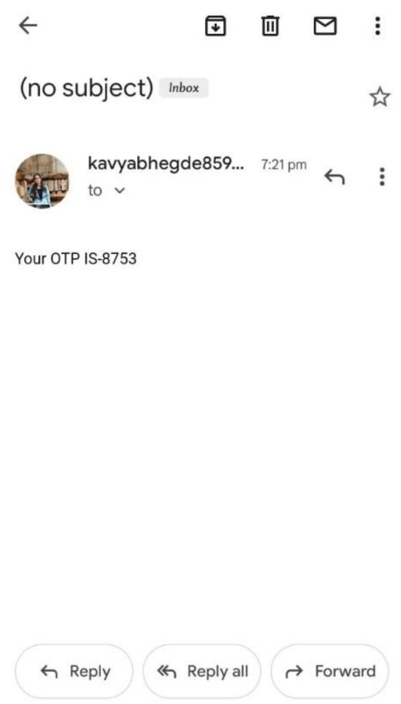
useremail vinodapatil28@gmail.com

Forgotpassword

```
def forgotpass(request):
    if request.method=="POST":
        username=request.POST.get('t1')
        request.session['username']=username
        ucheck=UserLogin.objects.filter(username=username).count()
        if ucheck>=1:
            otp = random.randint(1111, 9999)
            OtpCode.objects.create(otp=otp, status='active')
            content = "Your OTP IS-" + str(otp)
            mail = smtplib.SMTP('smtp.gmail.com', 587)
            mail.ehlo()
            mail.starttls()
            mail.login('kavyabhegde859@gmail.com','ignhyaqclqormgwx')
            mail. sendmail ('kavyabhegde859@gmail.com’, username,
content)
            return redirect('/otp')

        else:
            return render (request,'forgotpass.html',{'msg':'Invalid Username'})
    return render(request,'forgotpass.html')
```

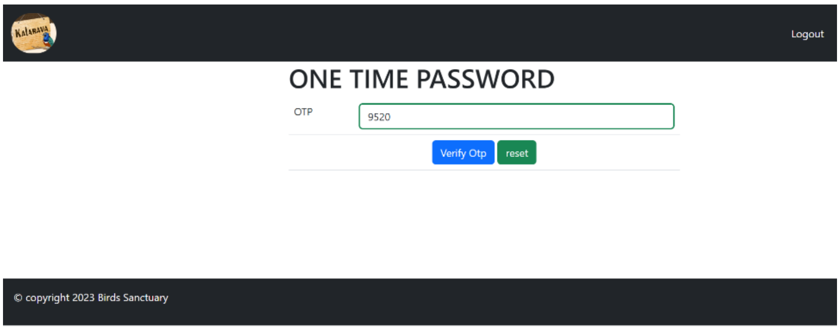
# Verification of Email With OTP:



(no subject) Inbox

kavyabhegde859... 7:21 pm
to

Your OTP IS-8753

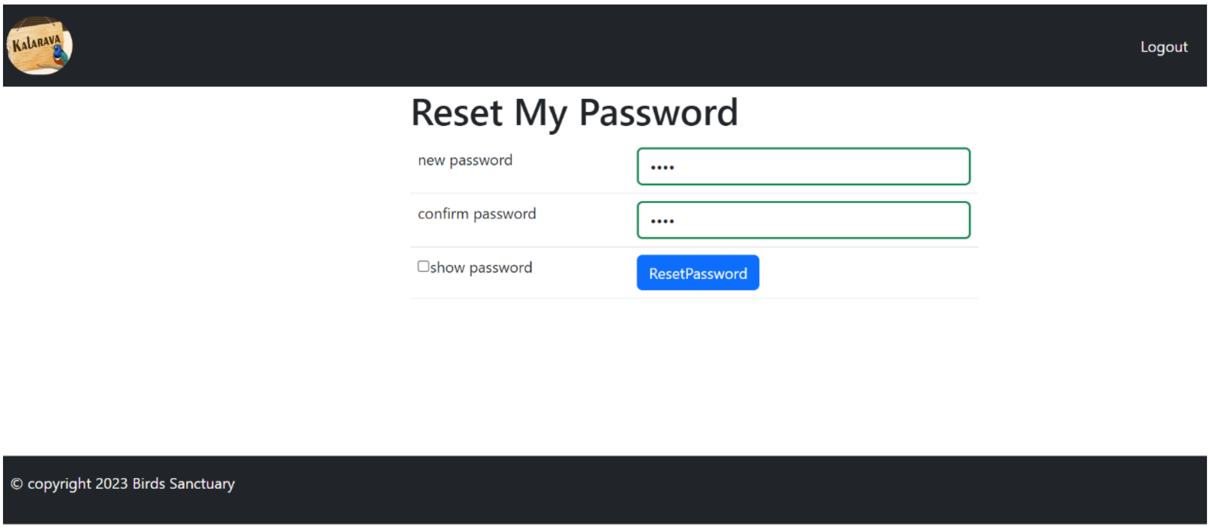


ONE TIME PASSWORD

OTP 9520

```
def otp(request):
if request. method=="POST":
otp=request.POST.get('t1')
ucheck=OtpCode.objects. filter(otp=otp). count ()
if ucheck>=1:
return redirect('/resetpass')
else:
return render (request,'otp.html',{'msg':'Invalid OTP'})
return render(request,'otp.html')
```

## Reset of Password:



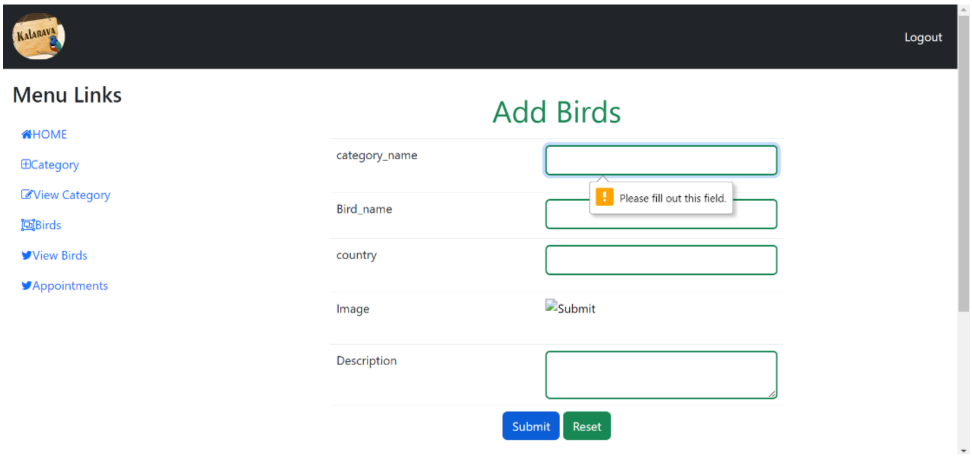
Reset My Password

| | |
|---|---|
| new password | •••• |
| confirm password | •••• |
| □show password | ResetPassword |

© copyright 2023 Birds Sanctuary

```
#Reset password
def resetpass(request):
    username=request.session['username']
    if request.method=="POST":
        newpass=request.POST.get('t1')
        confirmpass=request.POST.get('t2')
        if newpass==confirmpass:
            UserLogin.objects.filter(username=username).update(password=newpass)
            return redirect('login')
        else:
            return render (request,'resetpass.html',{'msg':'New password and confirm password must be same'})
    return render(request,'resetpass.html')
```

## Validation of the Forms:



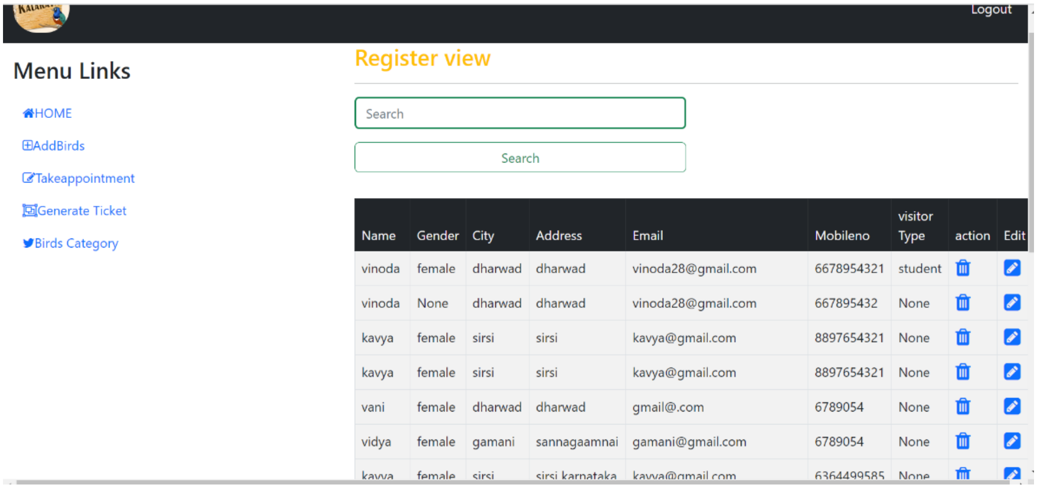
```
def add_bird_view(request):
      userdict=AddBirds.objects.all()
      return render(request,'add_bird_view.html',{'userdict':userdict})

{% include 'admin_menu.html' %}
<div class="container-fluid">
  <div class="row">
 <div class="col-lg-4">
<h3 class="mt-3"> Menu Links </h3>
  <nav class="navbar navbar-expand-*">
    <div class="container-fluid">
     <ul class="navbar-nav">
      <li class="nav-item">
           <a class="nav-link active text-primary" href="#"><i class="fa fa-home"></i>HOME</a>
     </li>
           <li class="nav-item">
                  <a class="nav-link text-primary" href="{% url 'add_birds' %}"><i class="fa fa-plus-square-o"></i>Add Birds</a>
            </li>
            <li class="nav-item">
                  <a class="nav-link text-primary" href="{% url 'take_app' %}"><i class="fa fa-pencil-square-o"></i>Take appointment</a>
            </li>
            <li class="nav-item">
```

```
              <a class="nav-link text-primary" href="{% url 'generate_tic' %}"><i
class="fa fa-object-group"></i>Generate Ticket</a>
            </li>
            <li class="nav-item">
             <a class="nav-link text-primary" href="{% url 'birds_category' %}"><i
class="fa fa-twitter"></i>Birds Category</a>
           </li>
           </ul>
          </div>
          </nav>
          </div>
         <div class="col-lg-8">
<body>
    <p style="color:red;text-align: center"> {{msg}} </p>
<form name="form1" method="post" action="" >
    {% csrf_token %}
    {% for i in userdict %}
<table class="table" style="width:600px;">
  <tr>
      <td colspan="2" class="text-success fs-1 text-center mt-3">Add
Birds </td>
    </tr>
  <tr>
    <td height="71"> category_name</td>
         <td><input type="text" name="t1" value="{{i.category_name}}"
class="form-control border border-2 border-success"required pattern="[A-
Za-z\s]+" title="Enter characters only"title="fill out this field"></td>
  </tr>
  <tr>
    <td width="261" height="61">Bird_name</td>
     <td width="304"><input type="text" name="t2" value="{{i.bird_name}}"
class="form-control border border-2 border-success"required pattern="[A-
Za-z\s]+" title="Enter characters only" pattern="[A-Za-z\s]+" title="Enter
characters only"title="fill out this field"></td>
  </tr>
  <tr>
    <td height="71">country</td>
    <td>
```

```
     <input type="text" name="t3” value="{{i. country}}" class="form-control
border border-2 border-success" title="Enter characters only"title="fill out
this field">
    </select></td>
  </tr>
  <tr>
    <td height="69">Image</td>
      <td><input type="image" border="0" value="{{i.image}}" name="t4"
src="file:///C|/Users/hp/AppData/Roaming/Macromedia/Dreamweaver%
20MX%202004/Configuration/Temp/FlashElements/ImageViewer/image">
</td>
  </tr>
  <tr>
    <td height="65">Description</td>
    <td><textarea name="t5"  class="form-control border border-2 border-
success" required pattern="[A-Za-z\s]+"
         title="fill out this field">{{i.description}}</textarea></td>
  </tr>
  <tr>
    <td height="100" colspan="4" align="center">
       <input type="submit" name="Submit" value="Submit" class='btn btn-
primary'>
         <input type="Reset" name="Reset" value="Reset" class='btn btn-
success'></td>
    </tr>
    {%  endfor %}
</table>
</form>
</div>
</div>
</div>
<div class="container-fluid bg-dark p-3 " style="margin-top:150px;">
<div class="row">
<p class="text-white"> &copy copyright 2023 Birds Sanctuary </p>
</div>
</div>
</body>
</html>
```

## View of the table:



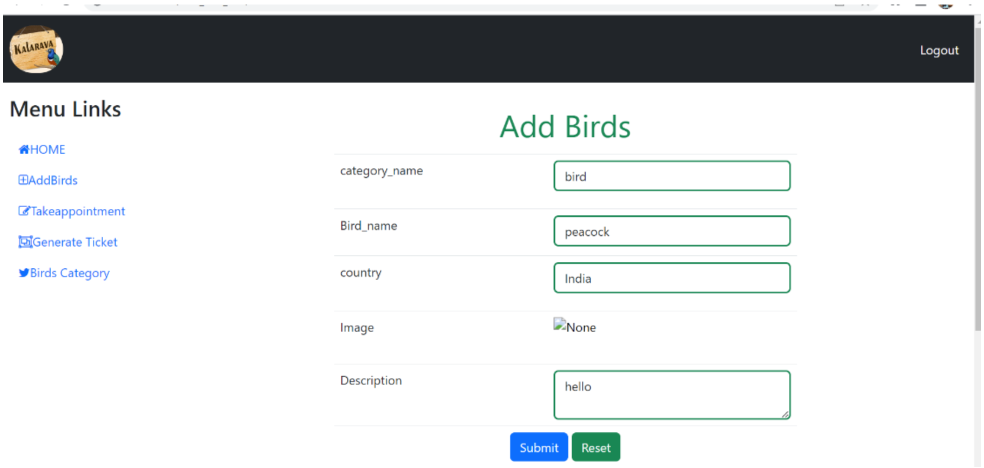
Register view

| Name | Gender | City | Address | Email | Mobileno | visitor Type | action | Edit |
|---|---|---|---|---|---|---|---|---|
| vinoda | female | dharwad | dharwad | vinoda28@gmail.com | 6678954321 | student | 🗑 | ✎ |
| vinoda | None | dharwad | dharwad | vinoda28@gmail.com | 667895432 | None | 🗑 | ✎ |
| kavya | female | sirsi | sirsi | kavya@gmail.com | 8897654321 | None | 🗑 | ✎ |
| kavya | female | sirsi | sirsi | kavya@gmail.com | 8897654321 | None | 🗑 | ✎ |
| vani | female | dharwad | dharwad | gmail@.com | 6789054 | None | 🗑 | ✎ |
| vidya | female | gamani | sannagaamnai | gamani@gmail.com | 6789054 | None | 🗑 | ✎ |
| kavya | female | sirsi | sirsi karnataka | kavya@gmail.com | 6364499585 | None | 🗑 | ✎ |

```
def reg_view(request):
      userdict=UserRegistration.objects.all()
      return render(request,'reg_view.html',{'userdict':userdict})
{% include 'admin_menu.html' %}
<div class="container-fluid">
  <div class="row">
  <div class="col-lg-4">
  <h3 class="mt-3"> Menu Links </h3>
   <nav class="navbar navbar-expand-*">
    <div class="container-fluid">
     <ul class="navbar-nav">
     <li class="nav-item">
  <a class="nav-link active text-primary" href="#"><i class="fa fa-
home"></i>HOME</a>
         </li>
           <li class="nav-item">
              <a class="nav-link text-primary" href="{% url 'add_birds' %}"><i
class="fa fa-plus-square-o"></i>AddBirds</a>
            </li>
            <li class="nav-item">
```

```
              <a class="nav-link text-primary" href="{% url 'take_app' %}"><i
class="fa fa-pencil-square-o"></i>Takeappointment</a>
            </li>
            <li class="nav-item">
              <a class="nav-link text-primary" href="{% url 'generate_tic' %}"><i
class="fa fa-object-group"></i>Generate Ticket</a>
            </li>
            <li class="nav-item">
              <a class="nav-link text-primary" href="{% url 'birds_category'
%}"><i class="fa fa-twitter"></i>Birds Category</a>
            </li>
           </ul>
        </div>
      </nav>
    </div>
    <div class="col-lg-8">
<form name="form1" method="post" action="">
  <h3 class="text-warning">Register view</h3>
  <hr>
  <form method="post">
    {% csrf_token %}
  <p> <input type="text" name="search" class="form-control  border-2
border border-success w-50" placeholder="Search">
    <input type="submit" class="btn btn-outline-success mt-3 mb-3 w-50"
value="Search"></p>
    </form>
    <table class="table table-bordered table-striped">
      <thead class="table-dark">
        <tr>
       <td >Name</td>
      <td>Gender</td>
      <td >City</td>
      <td >Address</td>
      <td >Email</td>
      <td >Mobileno</td>
      <td >visitor Type </td>
      <td >action</td>
      <td>Edit</td>
    </tr>
    </thead>
    <tbody>
    {% for i in userdict %}
    <tr>
      <td>{{i.name}} </td>
      <td> {{i. gender}} </td>
      <td> {{i. city}} </td>
      <td> {{i. address}} </td>
      <td> {{i. email}} </td>
      <td> {{i. mobile_no}} </td>
      <td> {{i. visitor type}}</td>
      <td><a href=" {% url 'register_del' i.id %}" onclick="return confirm ('are
u sure want to delete')">
        <i class="fa fa-trash fs-4 text-primary"></i></a></td>
      <td><a href="{% url 'register edit' i.id %}"><i class="fa fa-pencil-
square" style="font-size:24px"></i></a></td>
    </tr>
   </tbody>
  {% endfor %}
  </table>
</form>
</div>
</div>
</div>
<div class="container-fluid bg-dark p-3 " style="margin-top:150px;">
  <div class="row">
    <p class="text-white"> <marquee width="60%" direction="left"
height="50px">&copy copyright 2023 Birds Sanctuary </p>
  </div>
</div>
</body>
</html>
```

## Editing For the table:



Add Birds

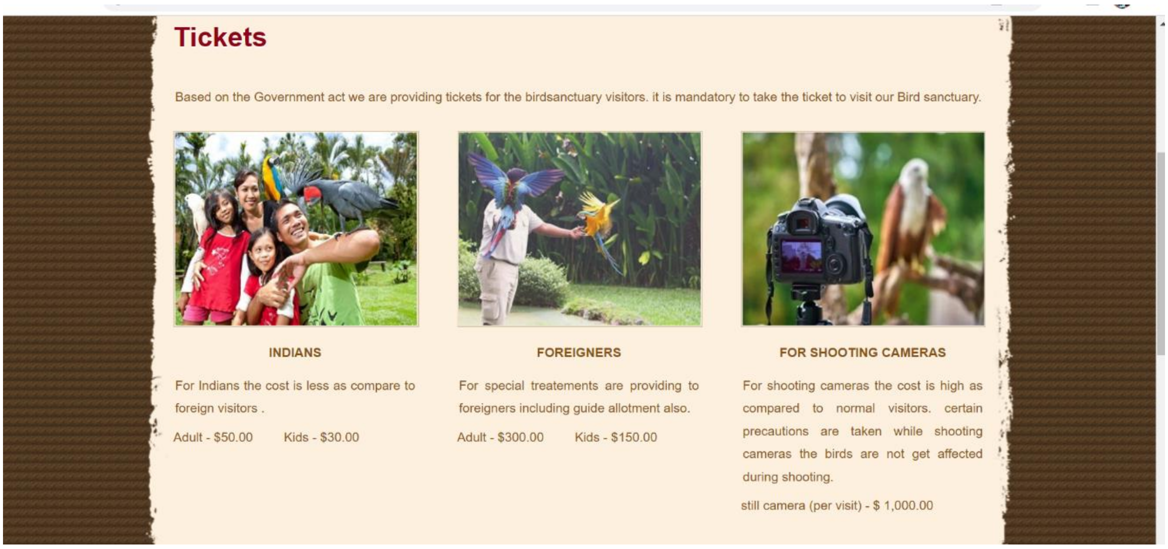
| | |
|---|---|
| category_name | bird |
| Bird_name | peacock |
| country | India |
| Image | None |
| Description | hello |

```
#Edit operation
def add_bird_edit(request):
    userdict=AddBirds. Objects. filter(id=pk). values ()
    if request.method=="POST":

        category_name= request.POST.get('t1')
        bird_name=request.POST.get('t2')
        country=request.POST.get('t3')
        image=request.POST.get('t4')
        description=request.POST.get('t5')


        AddBirds.objects.filter(id=pk).update(category_name=category_nam
e,bird_name=bird_name,country=country,image=image,
                                description=description)
        return redirect('/add_bird_view')
    return render(request,'add_bird_edit.html',{'userdict':userdict}
```

# Ticket allocation in the system:



## Tickets

Based on the Government act we are providing tickets for the birdsanctuary visitors. it is mandatory to take the ticket to visit our Bird sanctuary.

**INDIANS**

For Indians the cost is less as compare to foreign visitors .

Adult - $50.00 Kids - $30.00

**FOREIGNERS**

For special treatements are providing to foreigners including guide allotment also.

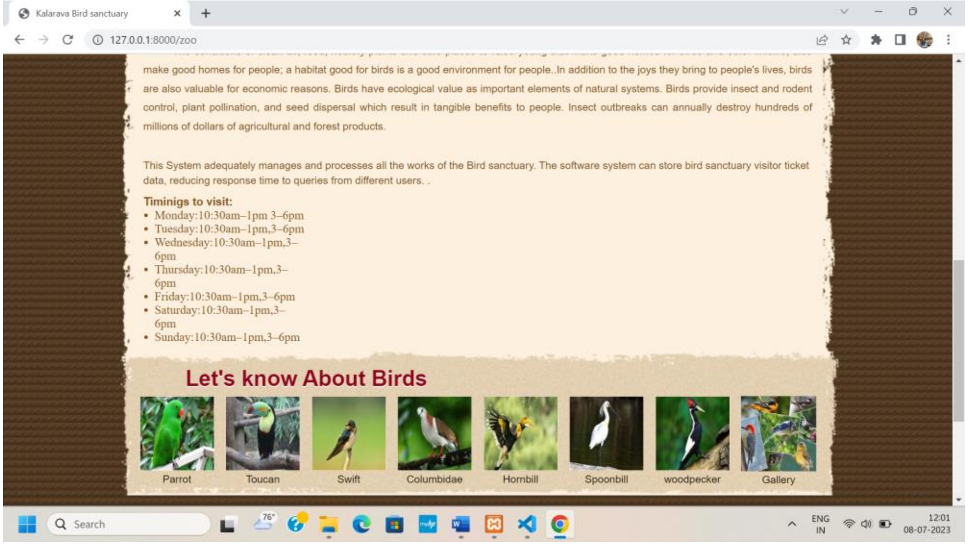
Adult - $300.00 Kids - $150.00

**FOR SHOOTING CAMERAS**

For shooting cameras the cost is high as compared to normal visitors. certain precautions are taken while shooting cameras the birds are not get affected during shooting.

still camera (per visit) - $ 1,000.00

# Time allocation for the visitors:



make good homes for people; a habitat good for birds is a good environment for people..In addition to the joys they bring to people's lives, birds are also valuable for economic reasons. Birds have ecological value as important elements of natural systems. Birds provide insect and rodent control, plant pollination, and seed dispersal which result in tangible benefits to people. Insect outbreaks can annually destroy hundreds of millions of dollars of agricultural and forest products.

This System adequately manages and processes all the works of the Bird sanctuary. The software system can store bird sanctuary visitor ticket data, reducing response time to queries from different users. .

**Timinigs to visit:**

- Monday:10:30am–1pm 3–6pm
- Tuesday:10:30am–1pm,3–6pm
- Wednesday:10:30am–1pm,3–6pm
- Thursday:10:30am–1pm,3–6pm
- Friday:10:30am–1pm,3–6pm
- Saturday:10:30am–1pm,3–6pm
- Sunday:10:30am–1pm,3–6pm

## Let's know About Birds

# SYSTEM TESTING AND RESULTING

## INTRODUCTION

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation. Testing is done to see if the entire feature provided by the module are performing satisfactorily and to ensure that the process of testing is as realistic as possible. Testing part is done to major phases via unit level and module level testing. The idea of testing the software in a phased manner is to identify and isolate the bugs for any easy correction. Both phases are over and results meet with the user needs.

## Test Approaches:

### Black box Testing:

Black box testing is done to find:

- Incorrect or missing functions.
- Interface Errors
- Errors in external database access
- Performance error.
- Initialization and termination error.

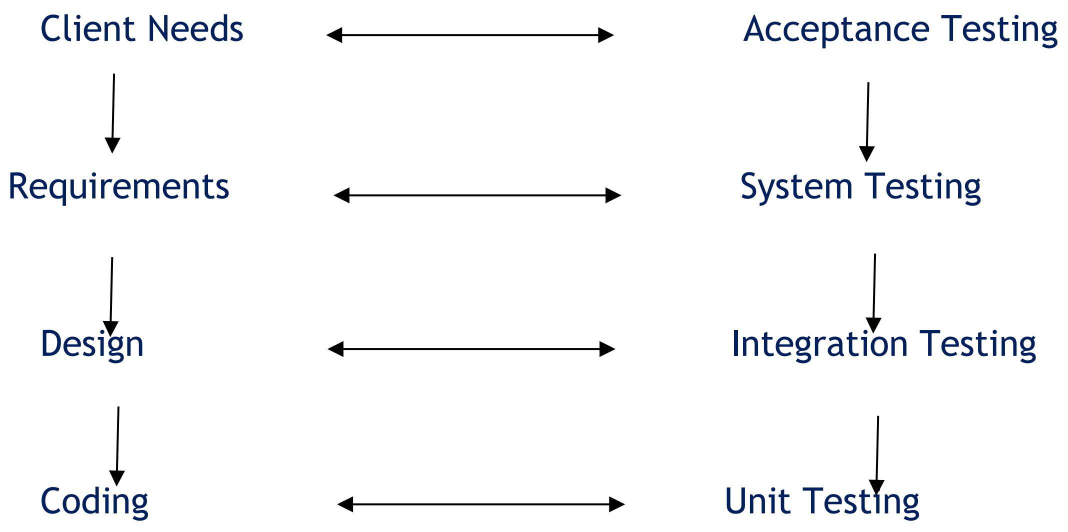
### White box Testing:

White box testing is done to find out:

- Check whether all independent paths within a module have been exercised at least once
- Exercise all logical decision on their true and false sides
- Execute all loops at their boundaries and within their bounds.
- Exercise the internal data structure to ensure their validity.
- Ensure whether all the possible validity checks and validity lookups have been provided to validate data entry.

### Testing Strategies:

### Levels of testing:



Client Needs ↔ Acceptance Testing

Requirements ↔ System Testing

Design ↔ Integration Testing

Coding ↔ Unit Testing

The different testing is carried out on the reflects the effectiveness and Efficiency of different phases of software development these test help to uncover the error in the corresponding phase.

- **Unit testing** is carried out to check the coding errors and program logic.

- **Integration testing** which also known as module testing uncovers the system design errors.
- **Performance testing** is performed to determine how fast some aspect of a system performs under a particular workload.
- **User Acceptance testing** is done to test whether the product developed needs the client needs and acceptable to him.

### Unit Testing:

Unit testing involves, checking all the modules in the system individually against the specification produced during the design of the module and for their performance. Unit testing also involves code produced in the coding phase and hence the internal logic of the program. Each module is tested for different test cases design to check each specific combination of conditions handled by the program. Error handlers are included in each module for each event trap and handled the errors.

Unit testing is done by inputting proper input s in each page and checking whether the data is in correct format as used to backend. Each Http Request is also checked whether each method is working properly to fulfill the requirement.

### Integration testing:

Integration testing takes as its input modules that have been checked out by unit testing, groups them in larger aggregates, applies tests defined in an Integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing. The purpose of integration testing is to verify functional, performance and reliability requirements placed on major design items.

### System Testing:

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic. As a rule, system testing takes, as its input, all of the "integrated" software components that have successfully passed integration testing and also the software system itself integrated with any applicable hardware system(s). The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together (called *assemblages*) or between any of the assemblages and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

### Security Testing:

This attempts to verify whether the protection mechanism is built into the system. In this testing the authentication of the users is checked and authorized users are allowed to access the database.

### Performance Testing:

Performance testing can serve different purpose. It can demonstrate that the system meets the performance criteria. It can compare two systems to find which performs better, or it can measure what parts of the system or workload cause the system to perform badly. In the diagnostic case, software engineers use tools such as profilers to measure what parts of a device or Software contributes most to the poor performance.

## Acceptance testing

Acceptance testing can mean one of two things:

1. A smoke test is used as an acceptance test prior to introducing a new build to the main testing process, i.e. before integration or regression.
2. Acceptance testing performed by the customer, often in their lab environment on their own hardware, is known as user acceptance testing (UAT). Acceptance testing may be performed as part of the hand-off process between any two phases of development.

## Test Cases and Results:

### Test case:

| Test Case | Input | Expected O/P | Actual O/P | Result |
|---|---|---|---|---|
| 1 | Valid Username and Password | It should display respective page according to user type. | Respective Home is displayed . | Passed |
| 2 | Invalid Username and Password | It should give appropriate error message saying "Enter proper User-n and Password" | Error message Displayed | Passed |
| 3 | Add/Update/ delete Member details | Add/Update/delete action is taken. | Add/Update/delete Member done successfully | Passed |
| 4 | User enters valid Username and password. | Respective Home is displayed | Respective Home is displayed . | Passed |

# CONCLUSION:

Software is said to have attained its objective only when it meets all the requirements of the user, further the user himself is the person to judge the success of the system.

Every attempt has been made to ensure that the system is fully functional and works effectively and efficiently. The system has been tested with simple data to cover all possible options and checked for all outputs. Since the system is flexible and modular, further modifications of this package can be easily incorporated.

## Future Enhancement of The Project:

- Management of Birds and species.
- In future Android application can be developed.
- Using some more tools on AI, based on Bird chirping Bird can be recognized easily.
- It provides the convenient way of communication to visitors through chatbots.

# BIBLIOGRAPHY:

## References:

1. Learning python by Mark Lutz 5th edition ,2013

2. Ian Summerville “Software Engineering” Pearson Education Ltd 6th edition 2004

3.An introduction to the python computer language and computer

Programming by Jason Cannon,2014

4. Learning web design by Jennifer Robbins’ 5th edition, 2008

5. Build a website with Django 3 by Nigel George 4th edition,2020

## Websites:

1. http://w3schools.com

2. www.tutorialspoint.com

3. https://youtu.be/TCZDSAuWiNo

4. https://getbootstrap.com/

5. https://youtu.be/QXeEoD0pB3E telusko