

A PROJECT REPORT ON
Pets Adoption System

Submitted by

Mr. Kartik Parab

in partial fulfillment for the award of
the degree of

BACHELOR OF SCIENCE

in

COMPUTER SCIENCE

under the guidance of

Mr. Pranam Kambli
Department of
Computer Science



**SHREE PANCHAM KHEMRAJ
MAHAVIDYALAYA SAWANTWADI**

(Sem V)

(2023– 2024)



**SHREE PANCHAM KHEMRAJ
MAHAVIDYALAYA SAWANTWADI**

**SAWANTWADI, 416510,
MAHARASTRA 2022-2023**

Department of Computer Science

CERTIFICATE

This is to certify that Mr. **Kartik M. Parab** of **T.Y.B.Sc. (Sem V)** class has satisfactorily completed the Project **Pets Adoption System**, to be submitted in the partial fulfillment for the award of **Bachelor of Science in Computer Science** during the academic year **2023– 2024**.

Date of Submission:

Project Guide

Head / Incharge

Department Computer Science

College Seal

Signature of Examiner

DECLARATION

I, **Kartik Mahendra Parab**, hereby declare that the project entitled “**PETS ADOPTION SYSTEM**” submitted in the partial fulfillment for the award of **Bachelor of Science in Computer Science** during the academic year **2023-2024** is my original work and the project has not formed the basis for the award of any degree, associateship, fellowship or any other similar titles.

Signature of the Student:

Place: Sawantwadi

Date: / /2023

ACKNOWLEDGEMENT

Special gratitude I give to my respected **Head Of Division Mrs. Vibha Gawande**, for allowing me to use the facilities available and also help me to coordinate my project Furthermore, I would also like to acknowledge with much appreciation the crucial role of faculty members on this occasion.

I extend my deepest appreciation to my esteemed guide, **Mr. Pranam Kambli** for providing me with the possibility to complete this project with the right guidance and advice.

Last but not least, I would like to thank friends who help me to assemble the parts and gave a suggestion about the project.

PROJECT TITLE

PET ADOPTION SYSTEM

ABSTRACT

This is a project for adoption of dogs and how we can improve the condition. Though there are animal shelters but some of them are cruel towards stray animals. In our project we also have Meet and Play where people can come on a particular day to visit and spend time with the animal. This entire project would involve the development of the Website using Html, CSS, PHP, JavaScript.

TABLE OF CONTENTS

1. INTRODUCTION

2. PROBLEM DEFINITION

3. OBJECTIVES

4. SCOPE AND LIMITATIONS

- SCOPE**
- LIMITATIONS**

5. REQUIREMENT ANALYSIS

- FUNCTIONAL REQUIREMENTS**
- NON-FUNCTIONAL REQUIREMENTS**

6. SPECIFICATIONS

- SOFTWARE SPECIFICATIONS**
- HARDWARE SPECIFICATIONS**

7. SYSTEM DEVELOPMENT ENVIRONMENT

8. DESIGN DOCUMENT

9. SYSTEM DESIGN

10. ACTIVITY DIAGRAM

11. UML DIAGRAM

12. USE CASE DIAGRAM

13. ER DIAGRAM

14. UML CLASS DIAGRAM

15. DATABASE DESIGN

16. SYSTEM TESTING

17. CODES

18. PROJECT EVALUATION – OUTPUT

19. CONCLUSION

20. REFERENCES

1. INTRODUCTION

Animal adoption is the process whereby a person brings an animal, mostly cats and dogs, to their own care. These animals that are either lost or abandoned are taken by the animal control to be kept in animal shelters. Pets that are kept in the shelters for a long time are euthanized to reduce the number of overpopulation of unwanted animals.

The inspiration of this project is a web-based adoption site for people to be more aware of stray animals waiting to be adopted around. The website is accessible for users and administrators where users can register themselves into the system to view the list of animals waiting to be adopted along with each animal's records such as their (estimated) age, gender, and their description. The administrators are the main authority of the system and they are responsible for adding, editing, updating, and deleting information if necessary.

2. PROBLEM DEFINITION

The current methods accustomed adopt or trying to find Dog is inefficient and haphazard. People distributed flyers to public and spread the news to the people round the area where the Dog is lost. When people print out flyers, it results into waste of paper and money because it doesn't have any positive outcome. People also post stories on social media platforms like Instagram, Facebook. There are numerous scammers too who try and falsely claim the reward for returning the Dog to their rightful owner. to avoid wasting animals from cruelty a Prevention of Cruelty to Animals Act was passed in 1960, however there have been no strict laws therein act as animals can be used for lab experiments and entertainment. Besides all this, animals on the streets are harassed by people by throwing stones only for entertainment. The numbers of stray animals have increased gradually over the years as people abandon their Dogs. Though there are many kind hearted people put still a platform is required where people come to understand the sufferings of stray animals.

3. OBJECTIVES

The objectives of this system are :

The goal of this project is to develop a website for users to select animals to adopt conveniently. The main objectives of this website development can be defined as follows:

- To develop a system that provides functions to support users to view the animals in the shelters conveniently.
- To maintain records of animals in the shelters and the users information in a centralized database system.
- To inform users of their result after sending their request to adopt an animal.

4. SCOPE AND LIMITATIONS

Scope:

- Connect shelters & adopters.
- Provide resources & info.
- Increase adoption rates.
- Build a pet adoption community.

Limitations:

- Relies on shelters' participation.
- Faces technological limitations.
- Needs to address ethical considerations.
- Limited by geography & resources.

5. REQUIREMENT ANALYSIS

FUNCTIONAL REQUIREMENTS –

1. **Add Pets**-In this section, shelter add pets details.
2. **Edit Pets Details**- In this section, shelter update pets details.
3. **Search pets**- In this section, User search for pets.
4. **Profile** - In this section, user can update his/her profile.
5. **Change Password** - In this section, user and shelter can change his/her password.
6. **Logout** - Through this button, user can log out.

NON-FUNCTIONAL REQUIREMENTS –

1. **Usability** - There is a consistency in all the modules and webpages. To ease the navigation there is a back tab to provide access to previous page. There is proper instruction on each page.
2. **Reliability** - Each data record is stored on a well-built efficient database schema. There is no risk of data loss. The internal evaluation of data is well coded.
3. **Supportability** - The system is well built to support any machine. Maintainability of the system is easy.
4. **Performance** - In order to ease the accessibility, the types of expenses are given.
5. **Availability** - The system is available all the time, no time constraint.

6. SPECIFICATIONS

SOFTWARE SPECIFICATIONS –

Technology used	PHP
Database	MySQL
Server	Apache
Frontend	HTML,CSS,BootStrap
User Interface Design	HTML,JQUERY,JAVASCRIPT
Web Browser	Google Chrome
Software	XAMPP,Visual Studio

HARDWARE SPECIFICATIONS-

RAM	1 GB
Hard Disk	40 GB

7. SYSTEM DEVELOPMENT ENVIRONMENT

- HTML

WHAT IS HTML?



To publish information for global distribution, one needs a universal-understood language, a kind of publishing mother tongue that all computers may potentially understand. The publishing language used by the World Wide Web is HTML (**Hyper Text Markup Language**).

HTML is an acronym which stands for **Hyper Text Markup Language** which is used for creating web pages and web applications.

Hyper Text : Hyper Text simply means "Text within Text." A text has a link within it, is a hypertext. Whenever you click on a link which brings you to a new webpage, you have clicked on a hypertext. Hyper Text is a way to link two or more web pages (HTML documents) with each other.

Markup language : A markup language is a computer language that is used to apply layout and formatting conventions to a text document. Markup language makes text more interactive and dynamic. It can turn text into images, tables, links, etc.

Web Page : A web page is a document which is commonly written in HTML and translated by a web browser. A web page can be identified by entering an URL. A Web page can be of the static or dynamic type. With the help of HTML only, we can create static web pages.

Hence, HTML is a markup language which is used for creating attractive web pages with the help of styling, and which looks in a nice format on a web browser. An HTML document is made of many HTML tags and each HTML tag contains different content.

- **CSS**



Cascading Style Sheets, fondly referred to as **CSS**, is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page. It describes how a webpage should look: it prescribes colors, fonts, spacing, and much more. In short, you can make your website look however you want. CSS lets developers and designers define how it behaves, including how elements are positioned in the browser.

- **PHP**



The term PHP is an acronym for **PHP Hypertext Preprocessor**. PHP is a server-side scripting language designed specifically for web development. It is open-source which means it is free to download and use. It is very simple to learn and use. The files have the extension “.php”. PHP code is executed in the server.

It can be integrated with many databases such as Oracle, Microsoft SQL Server, MySQL, PostgreSQL, Sybase, and Informix. One of the main reasons behind this is that PHP can be easily embedded in HTML files and HTML codes can also be written in a PHP file.

- **JavaScript**



JavaScript is the world most popular lightweight, interpreted compiled programming language. It is also known as scripting language for web pages. It is well-known for the development of web pages, many non-browser environments also use it. JavaScript can be used for **Client-side** developments as well as **Server-side** developments. JavaScript can be added to your HTML file in two ways –

- **Internal JS** : We can add JavaScript directly to our HTML file by writing the code inside the `<script>` tag. The `<script>` tag can either be placed inside the `<head>` or the `<body>` tag according to the requirement.

- **External JS** : We can write JavaScript code in other file having an extension .js and then link this file inside the `<head>` tag of the HTML file in which we want to add this code.

- **JQuery**



jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

- **MySQL**



Structured Query Language is a computer language that we use to interact with a relational database. SQL is a tool for organizing, managing, and retrieving archived data from a computer database. When data needs to be retrieved from a database, SQL is used to make the request. The DBMS processes the SQL query retrieves the requested data and returns it to us. Rather, SQL statements describe how a collection of data should be organized or what data should be extracted or added to the database.

In common usage, SQL encompasses DDL and DML commands for create, updates, modified or other operations on database structure.

SQL uses :

Data definition: It is used to define the structure and organization of the stored data and relationships among the stored data items.

Data retrieval : SQL can also be used for data retrieval.

Data manipulation : If the user wants to add new data, remove data, or modifying in existing data then SQL provides this facility also.

Access control : SQL can be used

Features of SQL :

- o SQL is an interactive query language.
- o SQL is a database administration language.
- o SQL is a database programming language.
- o SQL is a client/server language.
- o SQL is a distributed database language.
- o SQL is a database gateway language.

➤ **RDBMS CONCEPTS**

I. DATA ABSTRACTION –

A major purpose of a database system is to provide users with an abstract view of the data. This system hides certain details of how the data is stored and maintained. However in order for the system to be usable, data must be retrieved efficiently. The efficiency lead to the design of complex data structure for the representation of data in the database. Certain complexity must be hidden from

the database system users. This accomplished by defining several levels of abstraction at which the database may be viewed.

I II. CLASSIFICATION OF DATABASE –

There are 3 types of database approaches given below –

a. Hierarchical Database :

In this type of model data is represented in simple tree structured. The record at the top of tree is known as root, the root may have any number of dependents. Each of these may have any number of low level dependents and so on up to any number of levels. The disadvantages of the approach are that no independent record occurrence can exist without its superior.

b. Network Database :

In a Network database, data is represented by Network structure. In this approach record occurrence can have any number of superiors as well as any number of immediate dependents thus allow many to many correspondence directly than an hierarchical approach. The main disadvantage of the Network model is data representation is very complex resulting in complexity of the DML (Data Manipulation Language).

c. Relational Database :

The Relational model represents data and relationships among data by a collection of tables each of which has a number of columns with unique names.

8. DESIGN DOCUMENT

- The entire system is projected with a physical diagram which specifies the actual storage parameters that are physically necessary for any database to be stored on to the disk. The overall systems existential idea is derived from this diagram.
- The relation upon the system is structure through a conceptual ER-Diagram, which not only specifies the existential entities but also the standard relations through which the system exists and the cardinalities that are necessary for the system state to continue.
- The content level DFD is provided to have an idea of the functional inputs and outputs that are achieved through the system. The system depicts the input and output standards at the high level of the systems existence.

A DFD does not show a sequence of steps. A DFD only shows what the different process in a system is and what data flows between them.

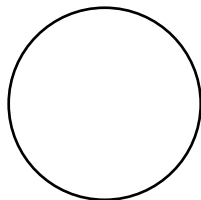
The following are some DFD symbols used in the project

External entities

Process : A transaction of information that resides within the bounds of the system to be module.



DATA STORE : A repository of data that is to be stored for use by one or more processes, may be as simple as buffer of queue or as a relational database



RULES FOR DFD :

- Fix the scope of the system by means of context diagrams.
- Organize the DFD so that the main sequence of the actions reads left to right and top to bottom.
- Identify all inputs and outputs.
- Identify and label each process internal to the system with rounded circles.
- A process is required for all the data transformation and transfers. Therefore, never connect a data store to a data source or the destinations or another data store with just a data flow arrow.
- Do not indicate hardware and ignore control information.
- Make sure the names of the processes accurately convey everything the process is done.
- There must not be unnamed process.
- Indicate external sources and destinations of the data, with squares.
- Number each occurrence of repeated external entities.
- Identify all data flows for each process step, except simple Record retrievals.
- Label data flow on each arrow.
- Use details flow on each arrow.
- Use the details flow arrow to indicate data movements.
- There can't be unnamed data flow.
- A data flow can't connect two external entities.

LEVELS OF DFD :

The complexity of the business system means that it is a responsible to represent the operations of any system of single data flow diagram. At the top level, an Overview of the different systems in an organization is shown by the way of context analysis diagram. When exploded into DFD They are represented by :

- LEVEL-0 : SYSTEM INPUT/OUTPUT
- LEVEL-1 : SUBSYSTEM LEVEL DATAFLOW FUNCTIONAL
- LEVEL-2 : FILE LEVEL DETAIL DATA FLOW.

The input and output data shown should be consistent from one level to the next.

LEVEL-0 : SYSTEM INPUT/OUTPUT LEVEL

A level-0 DFD describes the system-wide boundaries, dealing inputs to and outputs from the system and major processes. This diagram is similar to the combined user-level context diagram.

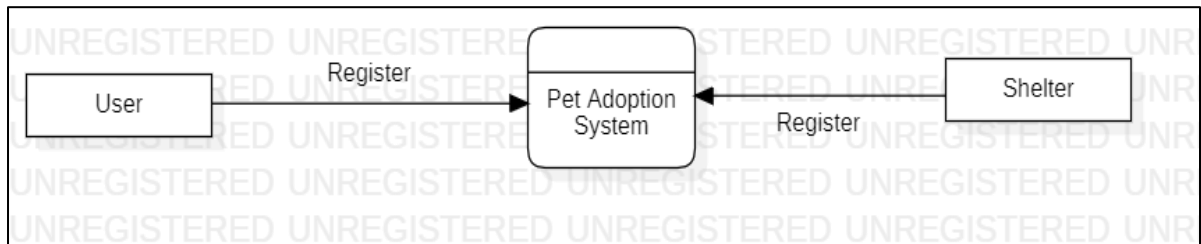
LEVEL-1 : SUBSYSTEM LEVEL DATA FLOW

A level-1 DFD describes the next level of details within the system, detailing the data flows between subsystems, which makeup the whole.

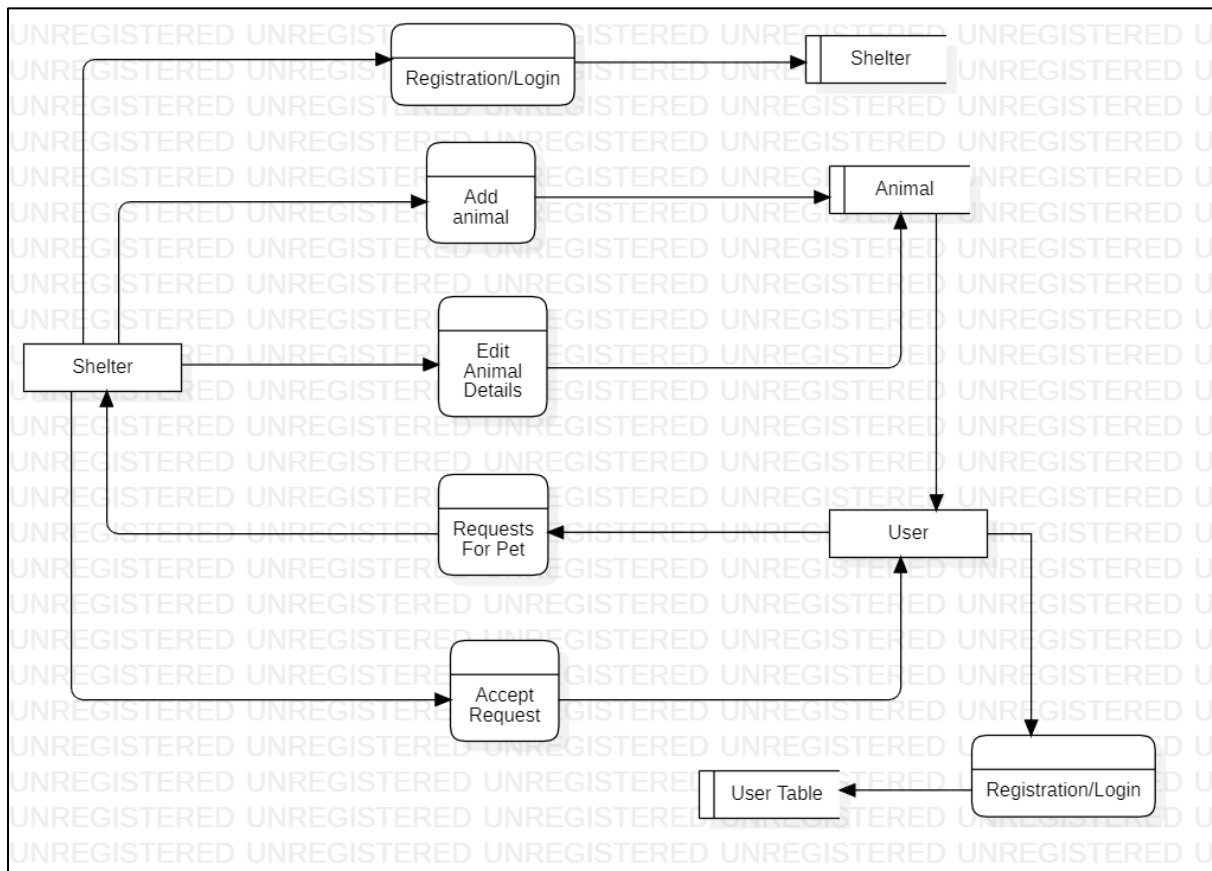
LEVEL-2 : FILE LEVEL DETAIL DATA FLOW

All the projects are feasible given unlimited resources and infinite time. It is both necessary and prudent to evaluate the feasibility of the project at the earliest possible time. Feasibility and the risk analysis are pertained in many ways. If project risk is great.

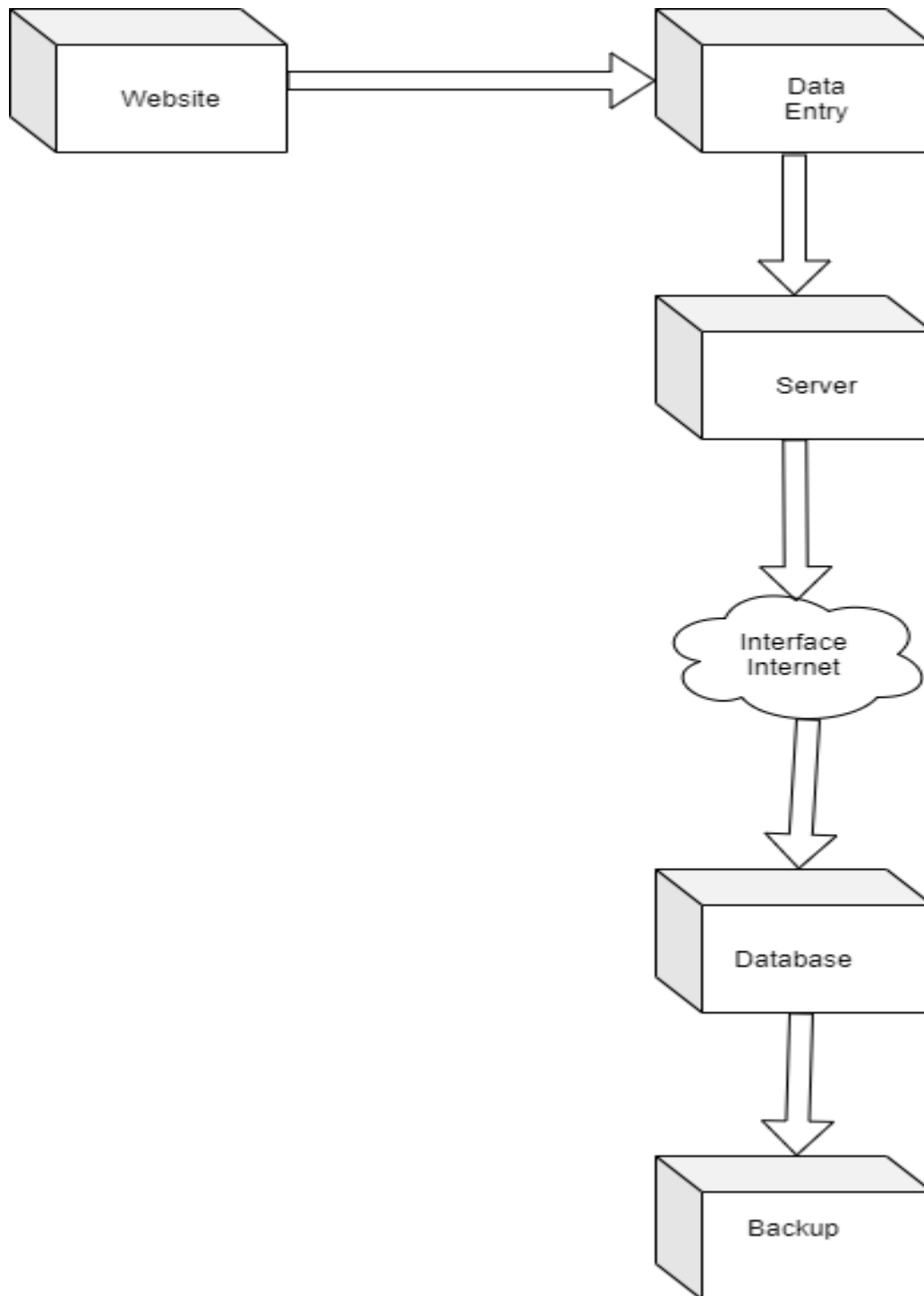
ZERO LEVEL DATAFLOW DIAGRAM



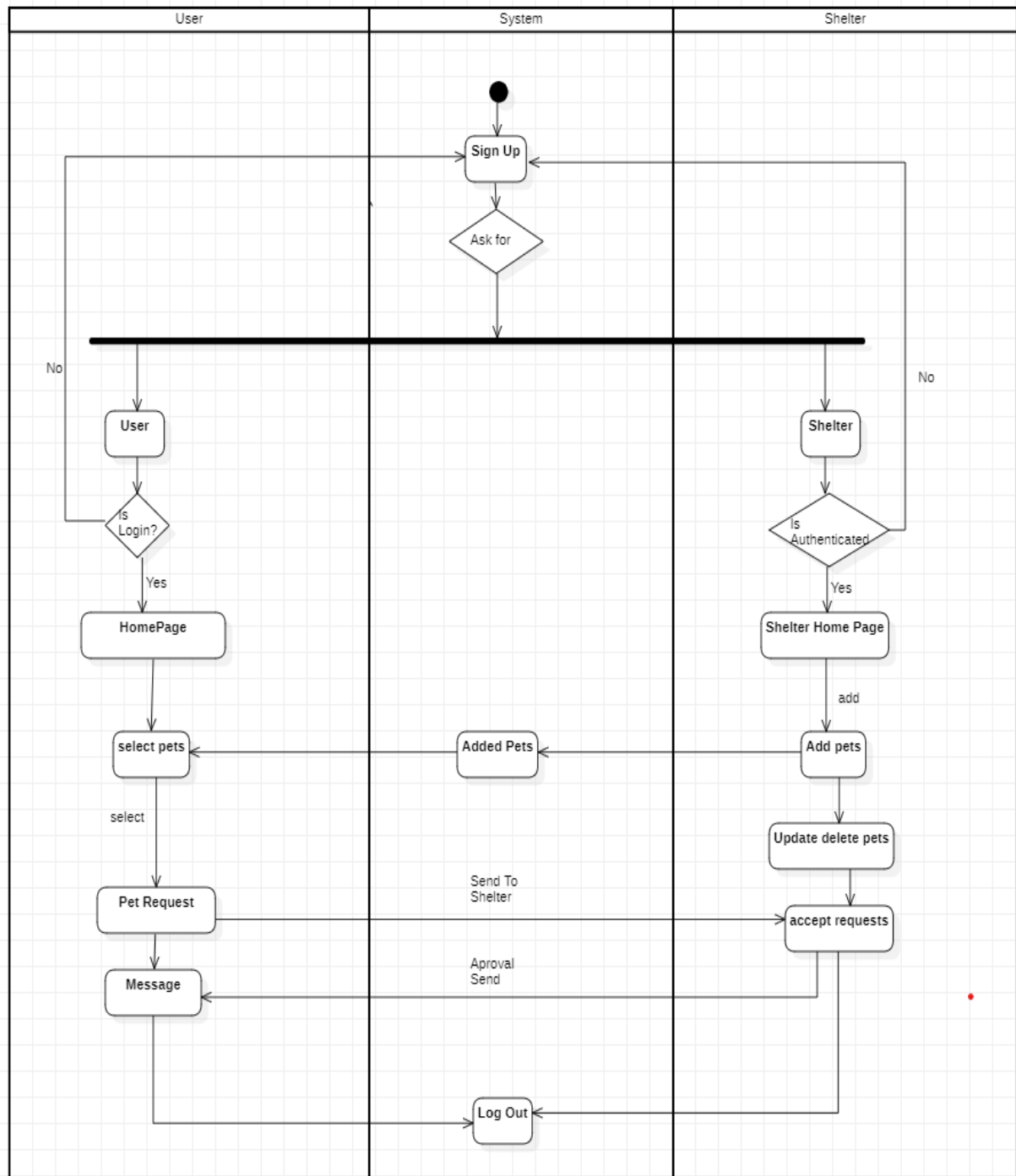
ONE LEVEL DATAFLOW DIAGRAM



9. SYSTEM DESIGN



10. ACTIVITY DIAGRAM



11. Unified Modeling Language Diagrams (UML)

The unified modeling language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

User Model View

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

Structural model view

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

Behavioral Model View

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

Implementation Model View

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

Environmental Model View

- In this the structural and behavioral aspects of the environment in which the system is to be implemented are represented.

UML is specifically constructed through two different domains they are :

UML Analysis modeling, which focuses on the user model and structural model views of the system?

UML design modeling, which focuses on the behavioral modeling, implementation modeling and environmental model views.

12. USE CASE DIAGRAM

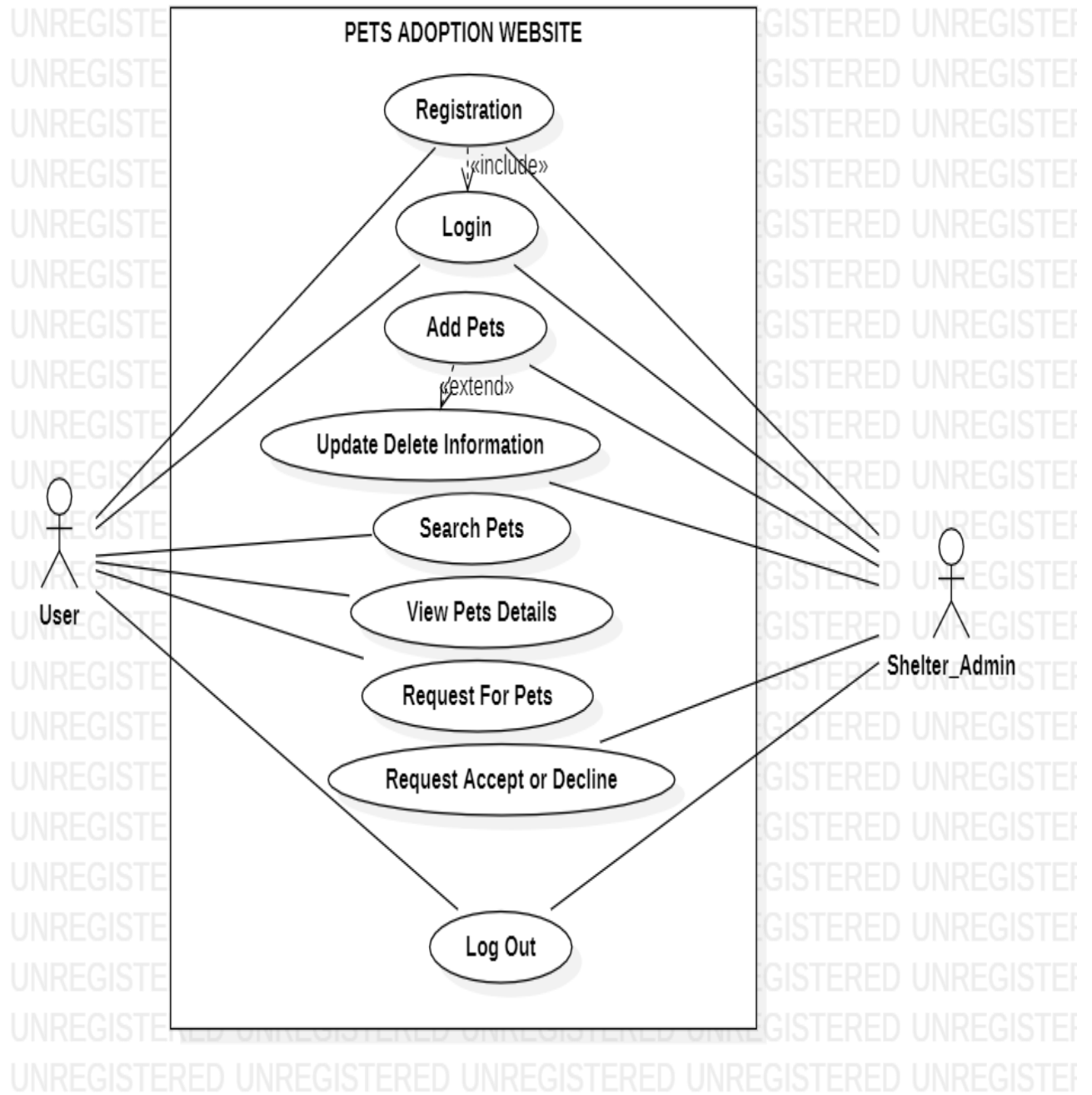
Use case diagrams model behavior within a system and helps the developers understand of what the user require. The stick man represents what's called an actor.

Use case diagram can be useful for getting an overall view of the system and clarifying who can do and more importantly what they can't do.

Use case diagram consists of use cases and actors and shows the interaction between the use case and actors.

- The purpose is to show the interactions between the use case and actor.
- To represent the system requirements from user's perspective.
- An actor could be the end-user of the system or an external system.

Use Case Diagram



13. ER DIAGRAM

The Entity-Relationship (ER) model was originally proposed by Peter in 1976 [Chen76] as a way to unify the network and relational database views. Simply stated the ER model is a conceptual data model that views the real world as entities and relationships. A basic component of the model is the Entity-Relationship diagram which is used to visually represent data objects. Since Chen wrote his paper the model has been extended and today it is commonly used for database design for the database designer, the utility of the ER model is

- It maps well to the relational model. The constructs used in the ER model can easily be transformed into relational tables.
- It is simple and easy to understand with a minimum of training. Therefore, the model can be used by the database designer to communicate the design to the end user.
- In addition, the model can be used as a design plan by the database developer to implement a data model in specific database management software.

E-R (Entity-Relationship) Diagram is used to represents the relationship between entities in the table.

The symbols used in E-R diagrams are :

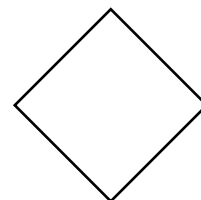
PURPOSE

SYMBOL

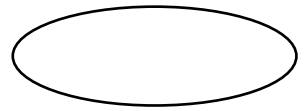
Represents Entity Sets



Represents Relationship Sets



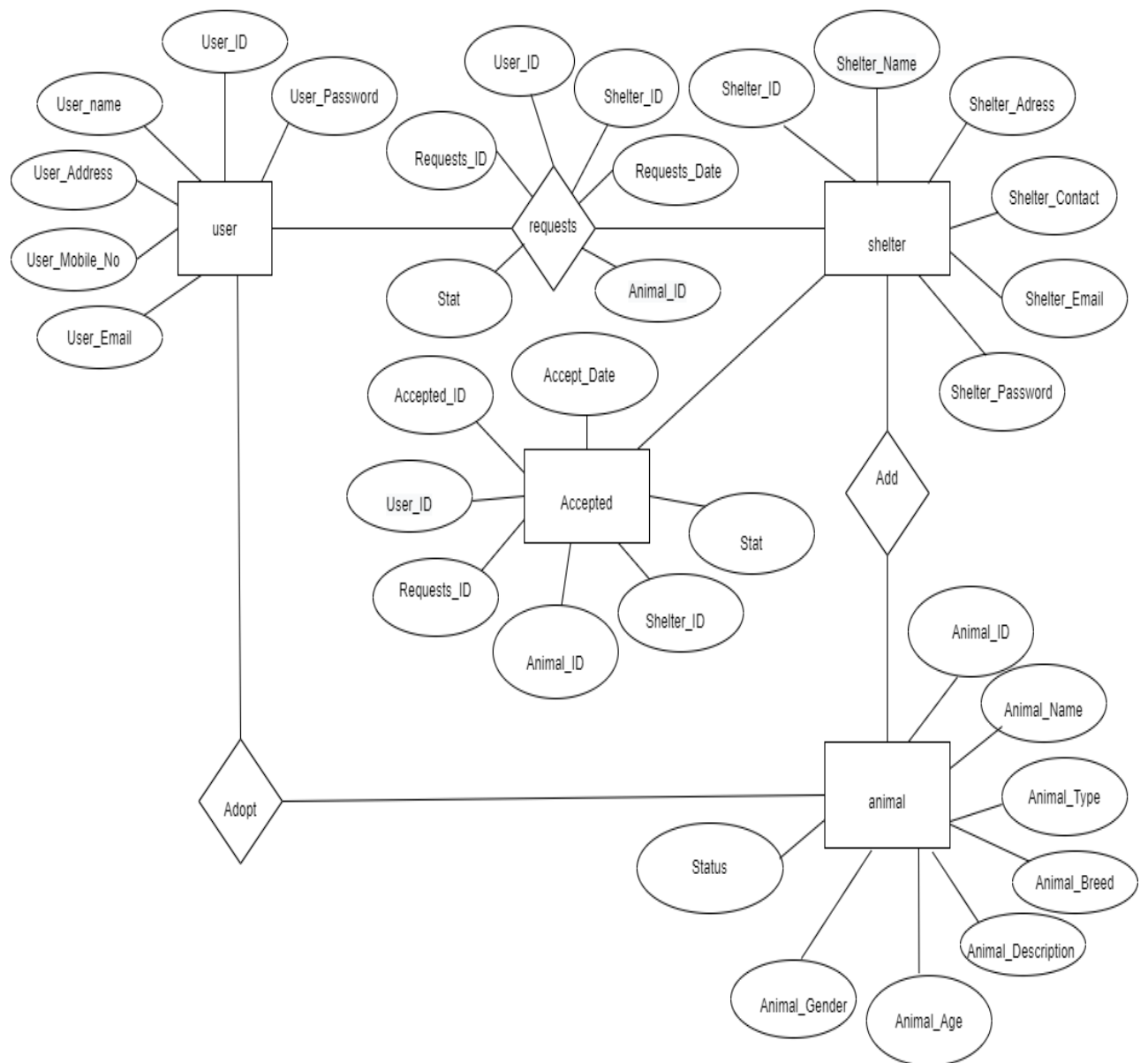
Represents Attributes



Line Represents Flow



ER DIAGRAM

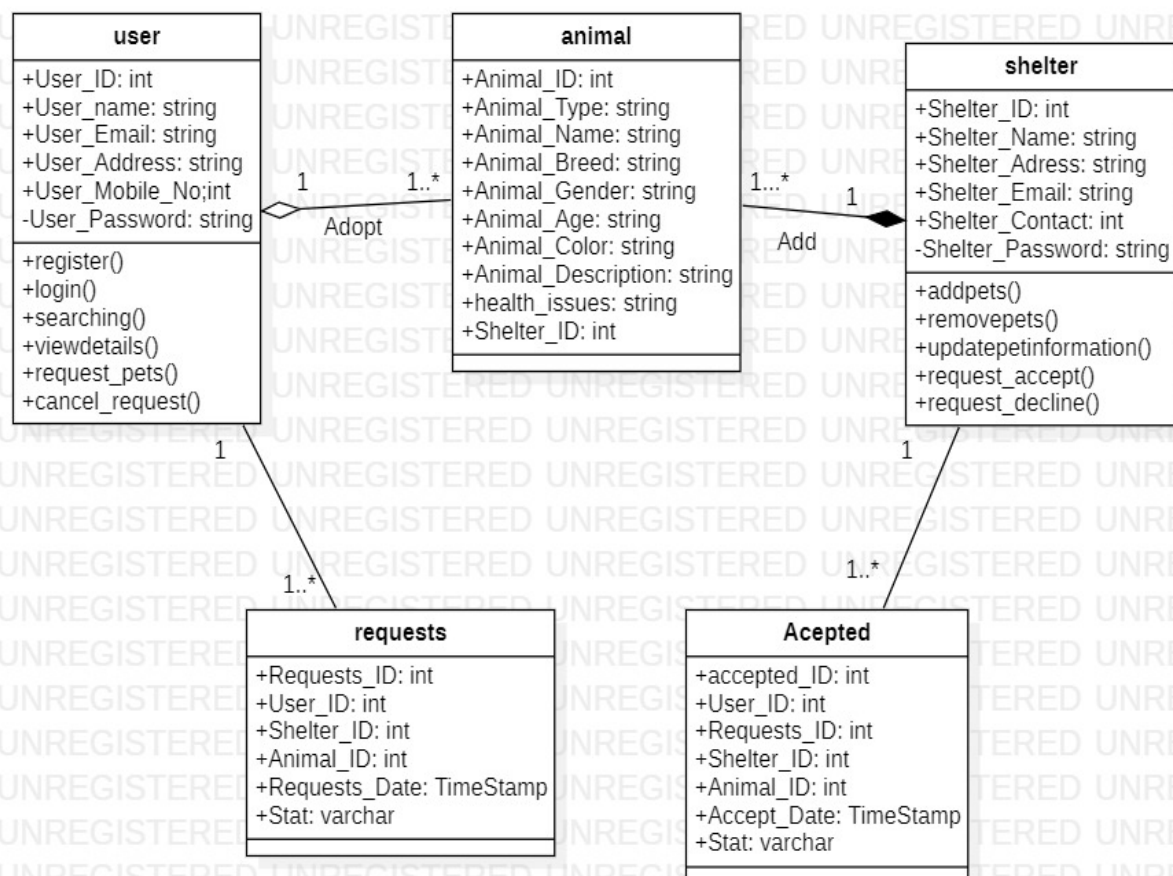


14. UML CLASS DIAGRAM

A description of set of objects that share the same attributes operations, relationships, and semantics. The class diagram depicts a static view of an application. It represents the types of objects residing in the system and the relationships between them. A class consists of its objects, and also it may inherit from other classes. A class diagram is used to visualize, describe, document various different aspects of the system, and also construct executable software code.

It shows the attributes, classes, functions, and relationships to give an overview of the software system. It constitutes class names, attributes, and functions in a separate compartment that helps in software development. Since it is a collection of classes, interfaces, associations, collaborations, and constraints, it is termed as a structural diagram.

CLASS DIAGRAM



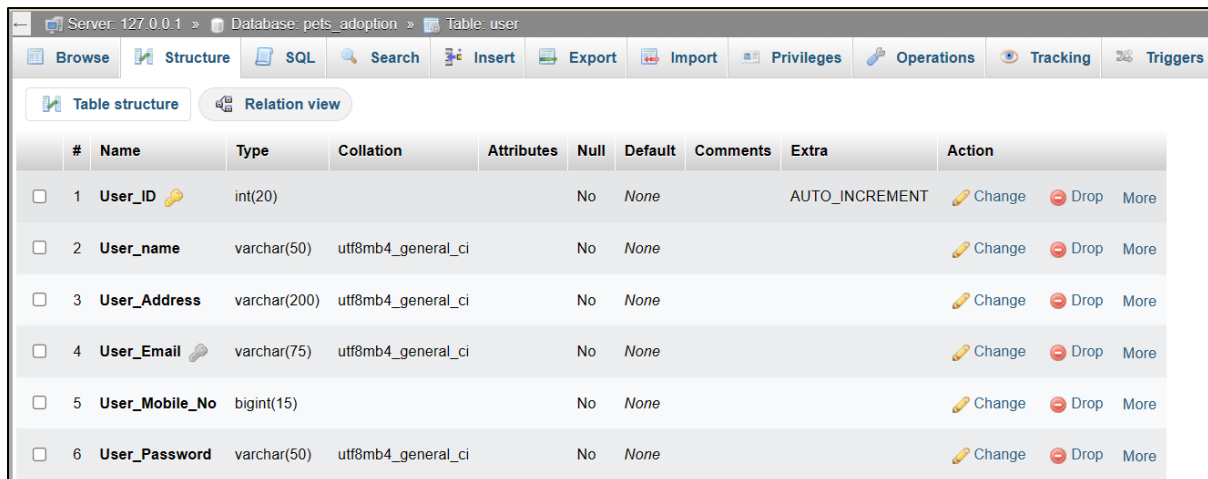
15. DATABASE DESIGN

The data in the system has to be stored and retrieved from database. Designing the database is part of system design. Data elements and data structures to be stored have been identified at analysis stage. They are structured and put together to design the data storage and retrieval system.

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make database access easy, quick, inexpensive and flexible for the user. Relationships are established between the data items and unnecessary data items are removed. Normalization is done to get an internal consistency of data and to have minimum redundancy and maximum stability. This ensures minimizing data storage required, minimizing chances of data inconsistencies and optimizing for updates. The MS Access database has been chosen for developing the relevant databases.

Daily Expense Tracking System (dets) contains 5 MySQL tables

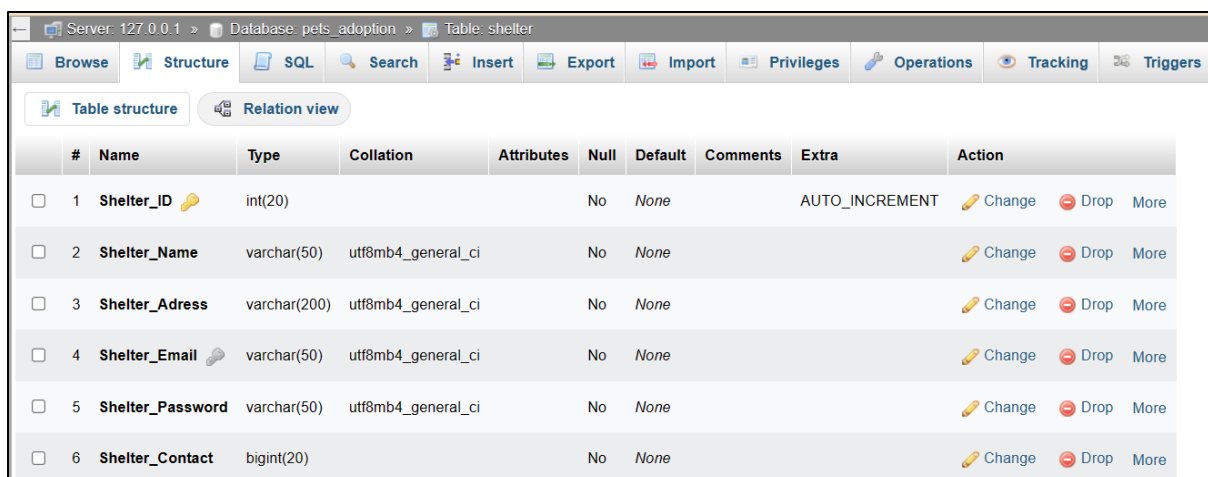
user table Structure : This table store the login and personal details of user.



The screenshot shows the MySQL Table structure for the 'user' table. The table has 6 columns: User_ID, User_name, User_Address, User_Email, User_Mobile_No, and User_Password. User_ID is the primary key and is set to AUTO_INCREMENT. All columns are of type VARCHAR or INT, with varying lengths and collations. The 'Action' column provides links to Change, Drop, and More options for each column.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	User_ID	int(20)			No	None		AUTO_INCREMENT	Change Drop More
2	User_name	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
3	User_Address	varchar(200)	utf8mb4_general_ci		No	None			Change Drop More
4	User_Email	varchar(75)	utf8mb4_general_ci		No	None			Change Drop More
5	User_Mobile_No	bigint(15)			No	None			Change Drop More
6	User_Password	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More

shelter table Structure : This table store the login and shelter details.



The screenshot shows the MySQL Table structure for the 'shelter' table. The table has 6 columns: Shelter_ID, Shelter_Name, Shelter_Address, Shelter_Email, Shelter_Password, and Shelter_Contact. Shelter_ID is the primary key and is set to AUTO_INCREMENT. All columns are of type VARCHAR or INT, with varying lengths and collations. The 'Action' column provides links to Change, Drop, and More options for each column.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	Shelter_ID	int(20)			No	None		AUTO_INCREMENT	Change Drop More
2	Shelter_Name	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
3	Shelter_Address	varchar(200)	utf8mb4_general_ci		No	None			Change Drop More
4	Shelter_Email	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
5	Shelter_Password	varchar(50)	utf8mb4_general_ci		No	None			Change Drop More
6	Shelter_Contact	bigint(20)			No	None			Change Drop More

animal table Structure : This table store the animal details.

Server: 127.0.0.1 » Database: pets_adoption » Table: animal										
Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers										
Table structure Relation view										
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action	
<input type="checkbox"/>	1 Animal_ID 🔑	int(20)			No	None		AUTO_INCREMENT		
<input type="checkbox"/>	2 Animal_Name	varchar(50)	utf8mb4_general_ci		No	None				
<input type="checkbox"/>	3 Animal_Age	varchar(50)	utf8mb4_general_ci		No	None				
<input type="checkbox"/>	4 Animal_Breed	varchar(50)	utf8mb4_general_ci		No	None				
<input type="checkbox"/>	5 Animal_Color	varchar(50)	utf8mb4_general_ci		No	None				
<input type="checkbox"/>	6 Animal_Type	varchar(50)	utf8mb4_general_ci		No	None				
<input type="checkbox"/>	7 Animal_Gender	varchar(50)	utf8mb4_general_ci		No	None				
<input type="checkbox"/>	8 Animal_Description	varchar(200)	utf8mb4_general_ci		No	None				
<input type="checkbox"/>	9 Shelter_ID 🔑	int(20)			No	None				
<input type="checkbox"/>	10 Animal_img 🔑	varchar(100)	utf8mb4_general_ci		No	None				
<input type="checkbox"/>	11 Status	varchar(20)	utf8mb4_general_ci		No	None				

requests table Structure : This table store the requested animal details.

Server: 127.0.0.1 » Database: pets_adoption » Table: requests										
Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers										
Table structure Relation view										
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action	
<input type="checkbox"/>	1 Requests_ID 🔑	int(20)			No	None		AUTO_INCREMENT		
<input type="checkbox"/>	2 Shelter_ID 🔑	int(20)			No	None				
<input type="checkbox"/>	3 Animal_ID 🔑	int(20)			No	None				
<input type="checkbox"/>	4 User_ID 🔑	int(20)			No	None				
<input type="checkbox"/>	5 Stat	varchar(100)	utf8mb4_general_ci		No	None				
<input type="checkbox"/>	6 Requests_Date	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()		

☐ Check all
 With selected:
 Browse
 Change
 Drop
 Primary
 Unique
 Index
 Spatial
 Fulltext
 Add to central columns

accepted table Structure : This table store the adopted animal details.

Server: 127.0.0.1 » Database: pets_adoption » Table: accepted

BrowseStructureSQLSearchInsertExportImportPrivilegesOperationsTrackingTriggers

Table structureRelation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 accept_ID	int(100)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 Requests_ID	int(100)			No	None			Change Drop More
<input type="checkbox"/>	3 Animal_ID	int(100)			No	None			Change Drop More
<input type="checkbox"/>	4 Stat	varchar(100)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	5 User_ID	int(100)			No	None			Change Drop More
<input type="checkbox"/>	6 Shelter_ID	int(20)			No	None			Change Drop More
<input type="checkbox"/>	7 Accept_Date	timestamp			No	current_timestamp()		ON UPDATE CURRENT_TIMESTAMP()	Change Drop More

16. SYSTEM TESTING

SOFTWARE TESTING TECHNIQUES :

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, designing and coding.

TESTING OBJECTIVES :

- Testing is process of executing a program with the intent of finding an error.
- A good test case design is one that has a probability of finding an as yet undiscovered error.
- A successful test is one that uncovers an as yet undiscovered error.

These above objectives imply a dramatic change in view port.

Testing cannot show the absence of defects, it can only show that software errors are present.

There are three types of testing strategies –

- a. Unit test
- b. Integration test
- c. Performance test

Unit Testing :

Unit testing focuses verification efforts on the smallest unit of software design module. The unit test is always white box oriented. The tests that occur as part of unit testing are testing the module interface, examining the local data structures, testing the boundary conditions, execution all the independent paths and testing error-handling paths.

Integration Testing :

Integration testing is a systematic technique or construction the program structure while at the same time conducting tests to uncover errors associated with interfacing. Scope of testing summarizes the specific functional, performance, and internal design characteristics that are to be tested. It employs top-down testing and bottom-up testing methods for this case.

Performance Testing :

Timing for both read and update transactions should be gathered to determine whether system functions are being performed in an acceptable timeframe.

17. CODES

UserHomePage.php

```
<!DOCTYPE html>
<html lang="en">
<head>
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
4bw+/aepP/YC94hEpVNVgiZdgIC5+VKNBQNGCHeKRQN+PtmoHDEXuppvnDJzQIu9"
crossorigin="anonymous">
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
HwwvtgBNo3bZJLYd8oVXjrBZt8cqVSpeBNS5n7C8IVInixGAoxmnlMuBnhbgrkm"
crossorigin="anonymous"></script>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-
icons@1.10.5/font/bootstrap-icons.css">
  <link rel="stylesheet" href="page.css">
  <script type="text/javascript"
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/jquery@3.6.0/dist/jquery.min.js"></script>
  <script type="text/javascript"
src="https://www.google.com/maps/embed/v1/MAP_MODE?key=AIzaSyCZZa9WE4h-
FVKXQTVgAdhyAsTa9VUb3AM&libraries=places"></script>
  <title>user Page</title>
</head>
<?php
session_start();
?>
<body>
  <div class="container back"></div>
  <nav class="navbar bg-dark body-tertiary fixed-top">
    <div class="container-fluid">
      <?php
      $conn =new mysqli('localhost','root','','pets_adoption');
      $user_id=$_SESSION['userid'];
    ?>
  <?php
  $sql01="select * from accepted inner join requests on accepted.Requests_ID=
requests.Requests_ID inner join animal on accepted.Animal_ID=animal.Animal_ID where
accepted.User_ID=$user_id and accepted.Stat='Unseen' and
requests.Shelter_ID=animal.Shelter_ID";
$result01=mysqli_query($conn,$sql01);
```

```

?>
<div class="list-group">
</div>
</div>
</div>
    <div class="brand"> <a class="navbar-brand" href="#">
        <h4 class="text-light ">PETS ADOPTION</h4>
    </a></div><a class="list-group-item">
<div class="btn-group" role="group" aria-label="Basic example">
<form action="" method="post">
    <button type="submit" id="not" class="btn btn-outline-light lab" name="not"><i class="fa
fa-check-circle " aria-hidden="true"></i>&nbsp;Adopt&nbsp;&nbsp;<span><?php
    if($result01->num_rows>0){
        $count_active = mysqli_num_rows($result01);
        echo $count_active;}else{ }?></span> </button></form></a>
    <a class="list-group-item" href="Requested pets.php"><button type="button" class="btn
btn-outline-light lab"><i class="fa fa-bell-o" aria-hidden="true"></i>&nbsp;
Requested</button></a>
    <a class="list-group-item" href="userprofile.php">
<button type="button" id="ADD" name="user_edit" class="btn btn-outline-light lab" data-
bs-toggle="modal" data-bs-target="#exampleModal"><i class="fa fa-user-o" aria-
hidden="true"></i>&nbsp;Profile</button>
    </a></div>
<form class="d-flex" role="search">
    <input class="form-control me-2" type="text" placeholder="Search" aria-label="Search"
name="name" id="getName">
    <button class="btn btn-outline-success" type="submit">Search</button>
</form>
</div>
    <div class="log"><a href="logout.php"><button type="button" class="btn btn-outline-
primary"><i class="fa fa-sign-out" aria-hidden="true"></i>&nbsp;Log
Out</button></a></div>
</div>
</nav>
<?php
if($_SERVER['REQUEST_METHOD']=='POST'){
    if (isset($_POST['not'])) {
        header("Location: accepted.php");
        $sql7="UPDATE accepted SET Stat = 'Seen' WHERE accepted.User_ID=$user_id
AND accepted.Stat = 'Unseen';
        $res02=mysqli_query($conn,$sql7);
        exit;
    }
}
?>
<div class="pt-5 mt-5 mx-5">
    <a href="dogs.php"><button class="btn btn-outline-warning rounded-
pill" style="width:27rem" type="button">Dogs</button></a>

```

```

<a href="cats.php"><button class="btn btn-outline-warning rounded-pill"
style="width:27rem" type="button">Cats</button></a>
<a href="others.php"><button class="btn btn-outline-warning rounded-pill"
style="width:27rem" type="button">Others</button></a>
</div>
<div class="container py-3" id="showdata">
<div class="row">
<?php
$sql = "SELECT * FROM animal inner join shelter where
animal.Shelter_ID=shelter.Shelter_ID and animal.Status='Not Accepted'";
$res=mysqli_query($conn,$sql);
if($res->num_rows>0){
while( $record = mysqli_fetch_assoc($res) )?>
<div class="col-md-3 mt-1">
<form name="requests form" method="post">
<div class="card" style="width: 20rem;">
<div class="image">

</div>
<div class="card-body"style="height:15rem;" >
<p class="name"><strong><?php echo $record['Animal_Name'];?></strong></p>
<p class="gender">Gender:<?php echo $record['Animal_Gender'];?></p>
<p class="age">Age:<?php echo $record['Animal_Age'];?></p>
<p class="adress">Address:<?php echo $record['Shelter_Adress'];?> </p>
<input type="hidden" name="" id="color" value="<?php echo
$record['Animal_Color'];?>">
<input type="hidden" name="" id="breed" value="<?php echo
$record['Animal_Breed'];?>">
<input type="hidden" name="" id="type" value="<?php echo
$record['Animal_Type'];?>">
<input type="hidden" name="" id="desc" value="<?php echo
$record['Animal_Description'];?>">
<input type="hidden" name="animal_id" value="<?php echo
$record['Animal_ID'];?>">
<input type="hidden" name="shelter_id" value="<?php echo
$record['Shelter_ID'];?>"><input type="hidden" name="" id="contact" value="<?php echo
$record['Shelter_Contact'];?>">
<input type="hidden" id="shelter" value="<?php echo $record['Shelter_Name'];?>">
<input type="hidden" id="img" value="<?php echo $record['Animal_Img'];?>">
<button type="submit" class="btn butt btn-block" name="request">Adopt</button>
<button type="button" id="<?php echo $record['Animal_ID'];?>" onclick="return
viewdetails(<?php echo $record['Animal_ID'];?>)" class="btn butt btn-
block" name="view">view details</button>
</div>
</div>
</form>
</div>

```



```

        <?php } }
        else{echo "<center><h1>No Animal For Adoption</h1></center>";}??
    </div>
</div>
<div class="modal fade bd-example-modal-lg" tabindex="-1" role="dialog" aria-
labelledby="myLargeModalLabel" aria-hidden="true" style="width:25rem;">
    <div class="modal-dialog modal-lg modal-dialog-centered">
        <div class="modal-content">
            

            <p class="view_name">
            <div class="row">
                <div class="col-sm"><p class="view_gender"></p></div> <div class="col-sm"><p
class="view_age"></p></div></div>

                <div class="row">
                    <div class="col-sm"> <p class="view_color"></p></div><div class="col-sm"><p
class="view_breed"></p></div>
                    </div><div class="row"><div class="col-sm"> <p class="view_desc"></p></div><div
class="col-sm"><p class="view_shelter"></p></div></div>
                    <div class="row"><div class="col-sm"> <p class="view_address"></p></div><div
class="col-sm"><p class="view_contact"> </div></div>
                </div>
            </div>
        </div>
    </div>
</div>
<script>
    var Adress='adress';
    $(document).ready(function(){
        var autocomplete;
        autocomplete= new
        google.maps.places.Autocomplete((document.getElementById(adress)),{
            types:['geocode'],});
    });
</script>
<?php
if($_SERVER['REQUEST_METHOD']=='POST'){
    if (isset($_POST['request'])) {
        $animal_id=$_POST['animal_id'];
        $shelter_id=$_POST['shelter_id'];
        $user_id2=$_SESSION['userid'];
        $req="select * from requests where Animal_ID = '$animal_id' and User_ID='$user_id2'";
        $req1=mysqli_query($conn,$req);
        if($req1->num_rows>0){
            ?><div class="alert alert-warning alert-dismissible mt-5 fade fixed-top show"
role="alert">
                <span>Your Already Request for this animal</span>
                <button type="button" class="btn-close" data-bs-dismiss="alert" aria-
label="Close"></button>

```

```

</div> <?php exit;
}
else{
    $sql3="INSERT INTO requests ( Shelter_ID, Animal_ID, User_ID,Stat) VALUES
('$shelter_id', '$animal_id', '$user_id2','Unseen')";
    $res3=mysqli_query($conn,$sql3);
    exit;
}
}}
?>
<script>
function viewdetails(id){
    var id=id;
    var name=$("#"+id).parent(".card-body").children(".name").text();
    var type=$("#"+id).parent(".card-body").children("#type").val();
    var img=$("#"+id).parent(".card-body").children("#img").val();
    var gender=$("#"+id).parent(".card-body").children(".gender").text();
    var breed=$("#"+id).parent(".card-body").children("#breed").val();
    var color=$("#"+id).parent(".card-body").children("#color").val();
    var age=$("#"+id).parent(".card-body").children(".age").text();
    var desc=$("#"+id).parent(".card-body").children("#desc").val();
    var adress=$("#"+id).parent(".card-body").children(".adress").text();
    var contact=$("#"+id).parent(".card-body").children("#contact").val();
    var shelter=$("#"+id).parent(".card-body").children("#shelter").val();
    var img=$("#"+id).parent(".card-body").children("#img").val();
    $(".view_name").text("name:"+name);
    $(".view_age").text(age);
    $(".view_desc").text("desc:"+desc);
    $(".view_gender").text(gender);
    $(".view_address").text(adress);
    $(".view_breed").text("breed:"+breed);
    $(".view_contact").text("contact:"+contact);
    $(".view_color").text("color:"+color);
    $(".view_shelter").text("shelter_name:"+shelter);
    $("#myImg").attr('src','./img/'+img);
    $(".bd-example-modal-lg").modal("toggle");
}
</script>
<script>
$(document).ready(function(){
    $('#getName').on("keyup", function(){
        var getName = $(this).val();
        $.ajax({
            method:'POST',
            url:'search.php',
            data:{ name:getName },
            success:function(response)

```

```

        {
            $("#showdata").html(response);
        }
    });
});
});
</script>
</body>
</html>

```

Index.php

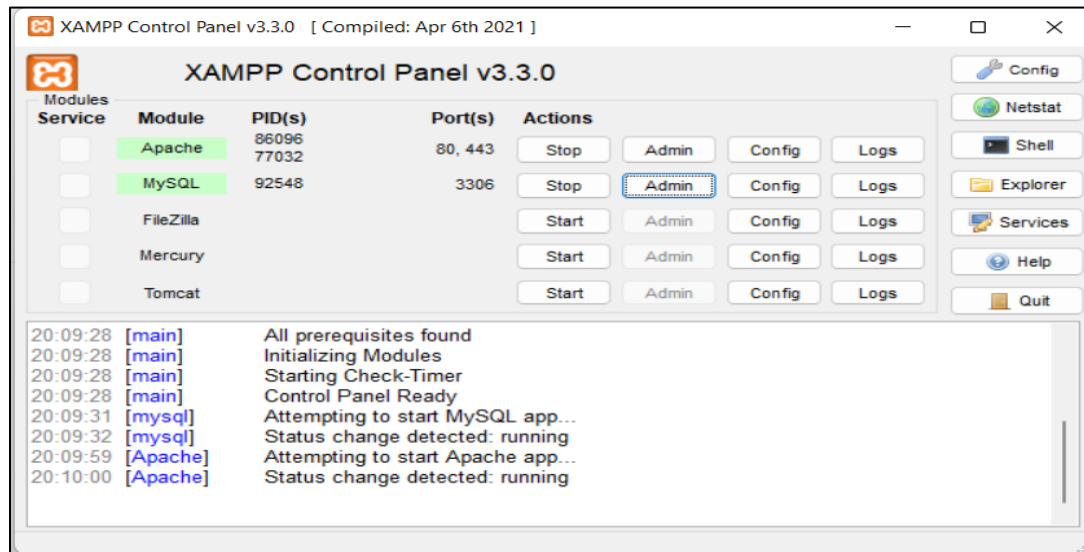
```

<html lang="en">
<head>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-
    HwwvtgBNo3bZJJLYd8oVXjrBZt8cqVSpeBNS5n7C8IVInixGAoxmnlMuBnhbgrkm"
    crossorigin="anonymous"></script>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.1/dist/css/bootstrap.min.css"
    rel="stylesheet" integrity="sha384-
    4bw+ /aepP/YC94hEpVNVgiZdgIC5+VKNBQNGCHeKRQN+PtmoHDEXuppvnDJzQIu9"
    crossorigin="anonymous">
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>index</title>
    <link rel="stylesheet" type="text/css" href="login.css">
</head>
<?php session_start(); ?>
<body>
<?php
$user_error=null;
if($_SERVER['REQUEST_METHOD']=='POST'){
$email=$_POST['email'];
$pass=$_POST['password'];
$conn =new mysqli('localhost','root','','pets_adoption');
// Check connection
if (!$conn) {
}else{
    $sql="select * from user where User_Email= '$email'";
    $res=mysqli_query($conn,$sql);
    if($res->num_rows>0){
        $data=$res->fetch_assoc();
        if($data['User_Password']===$pass){
            header("Location:userhome.php");
            $_SESSION['userid']=$data['User_ID'];
            $_SESSION['username']=$data['User_name'];
        }
    }
}
}

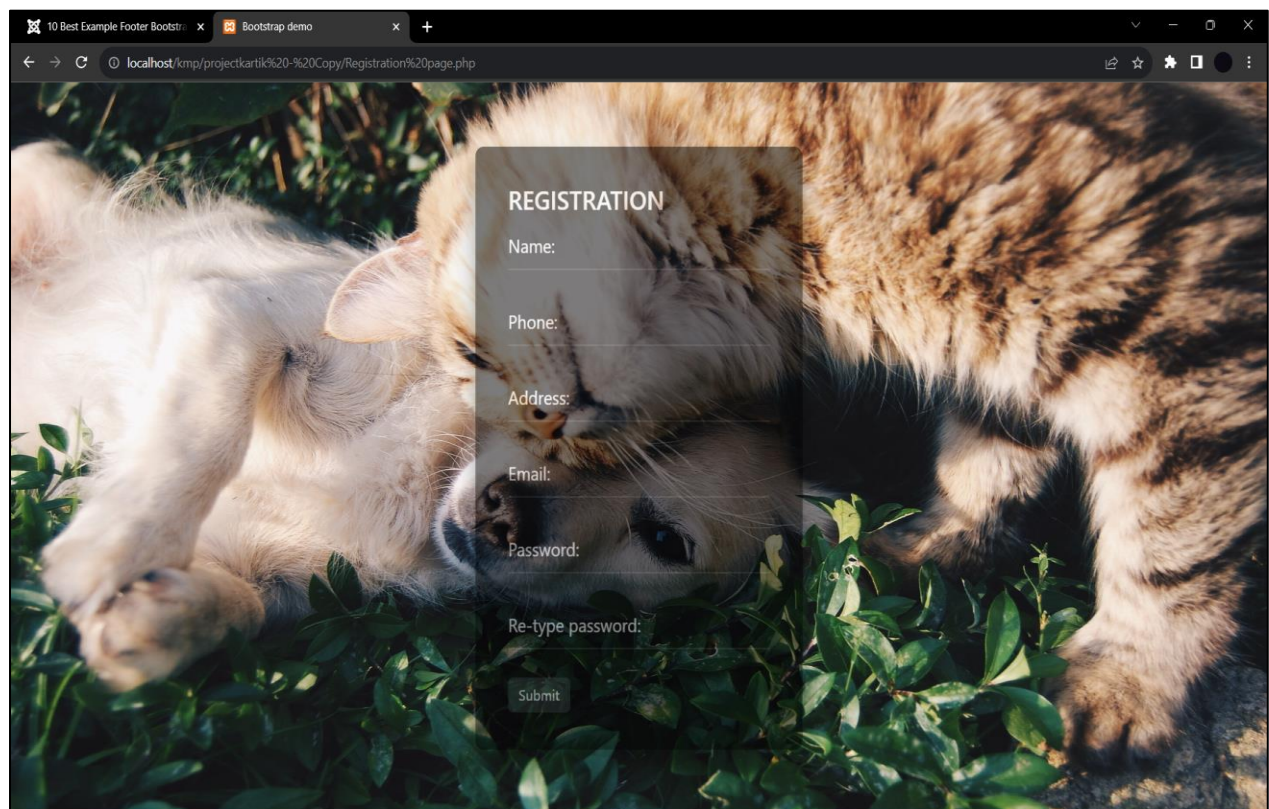
```

```
        else{
            $user_error="invalid password ";
        }
    }else{
        $user_error="invalid Email";
    }
}
}}
?>
<div class="wrapper">
    <form name="myform" onsubmit="return login ()" action="" method="post">
        <div class="single-div">
            <input type="email" name="email" id="email" required="">
            <label>Your Email</label>
            <span id="mail"></span>
        </div>
        <div class="single-div">
            <input type="password" name="password" id="password" required="" >
            <label>Your Password</label>
            <span id="error"><?php echo $user_error ; ?></span>
        </div>
        <div class="sub">
            <input type="submit" class="btn btn-dark">
        </div>
        <a href="Registration page.php">register as user? </a><br>
        <a href="shelter register.php">register as shelter?</a><br>
        <a href="shelter login.php">login as shelter?</a>
    </form>
</div>
</body>
<script src="login.js"></script>
</html>
```

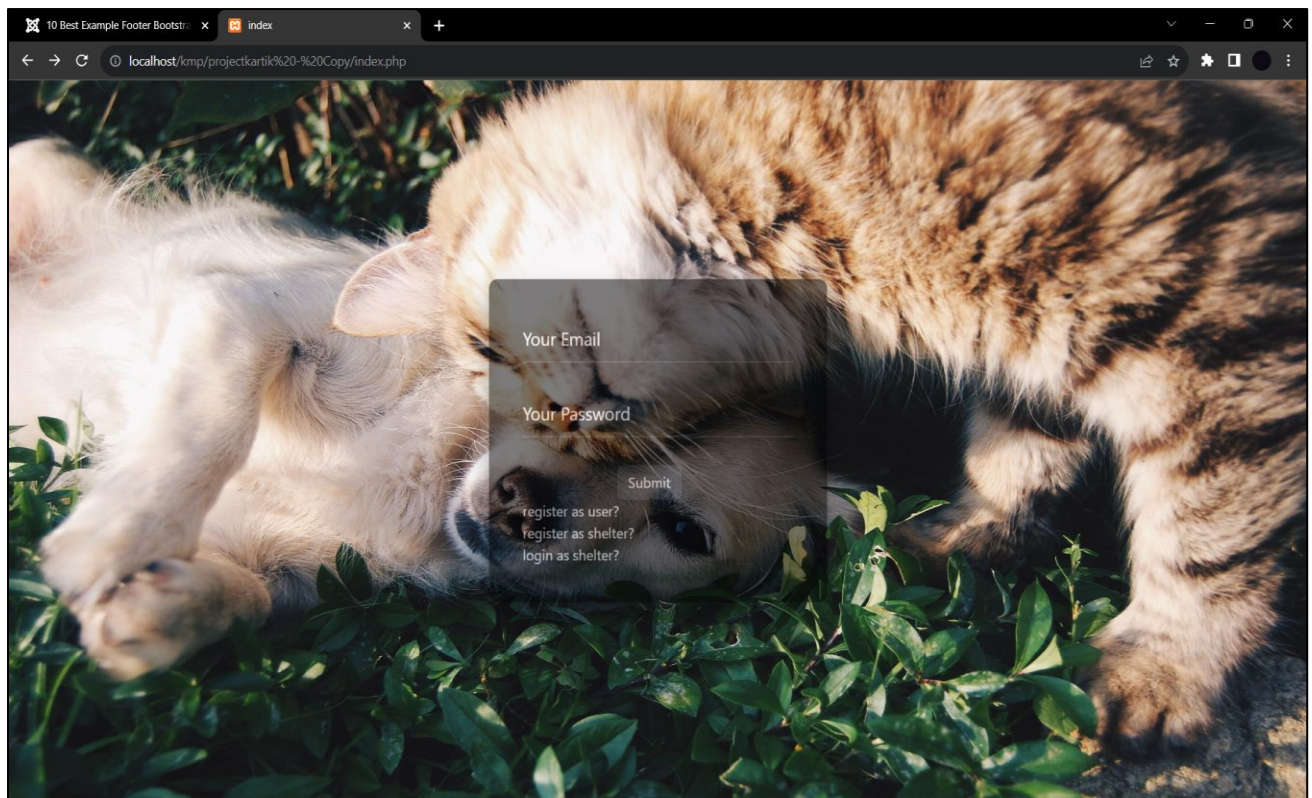
18. Project Evaluation – Output



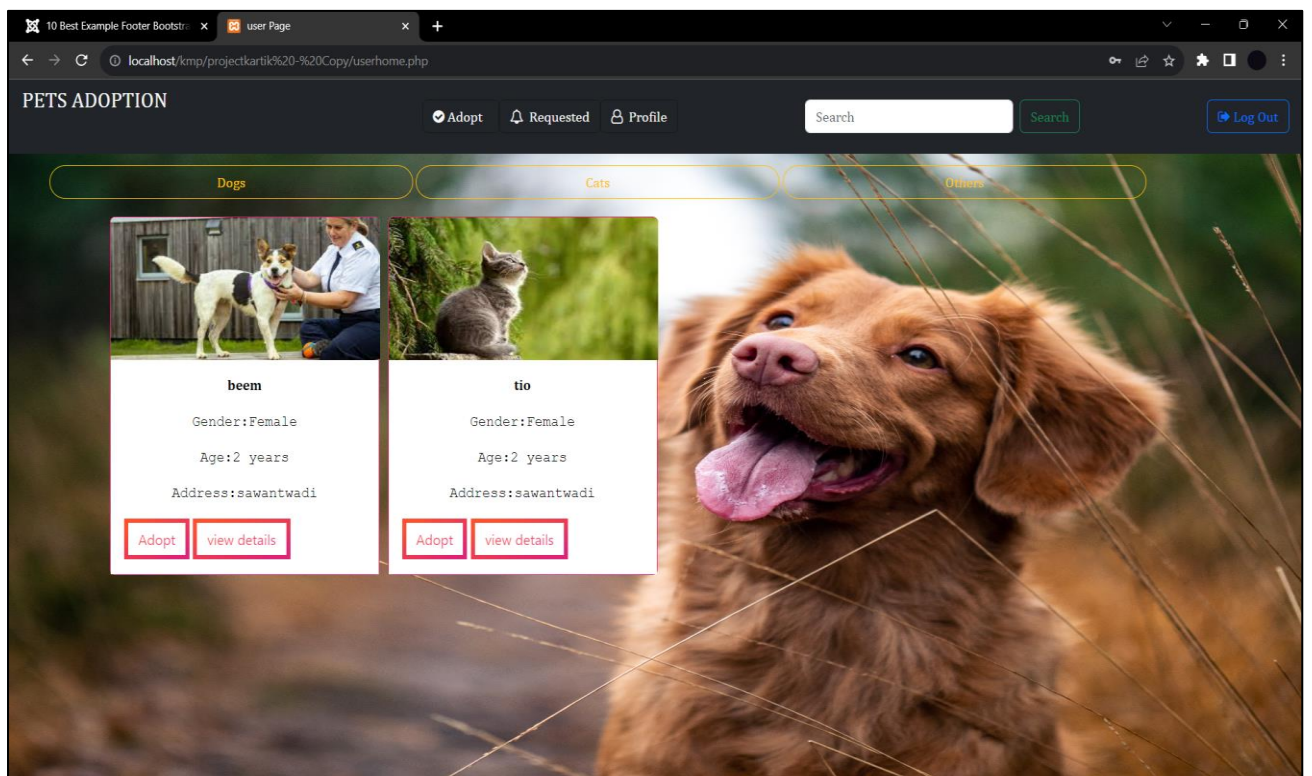
User Registration Page



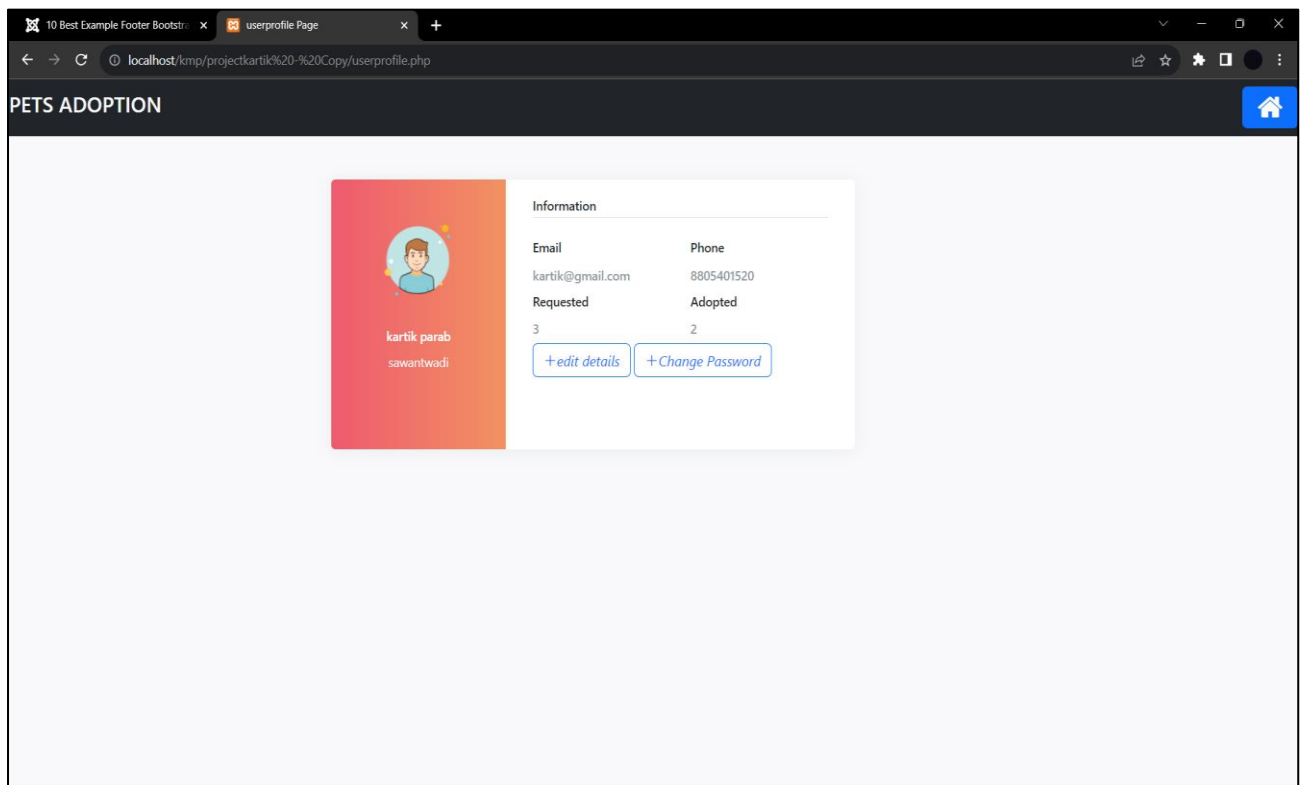
User Login



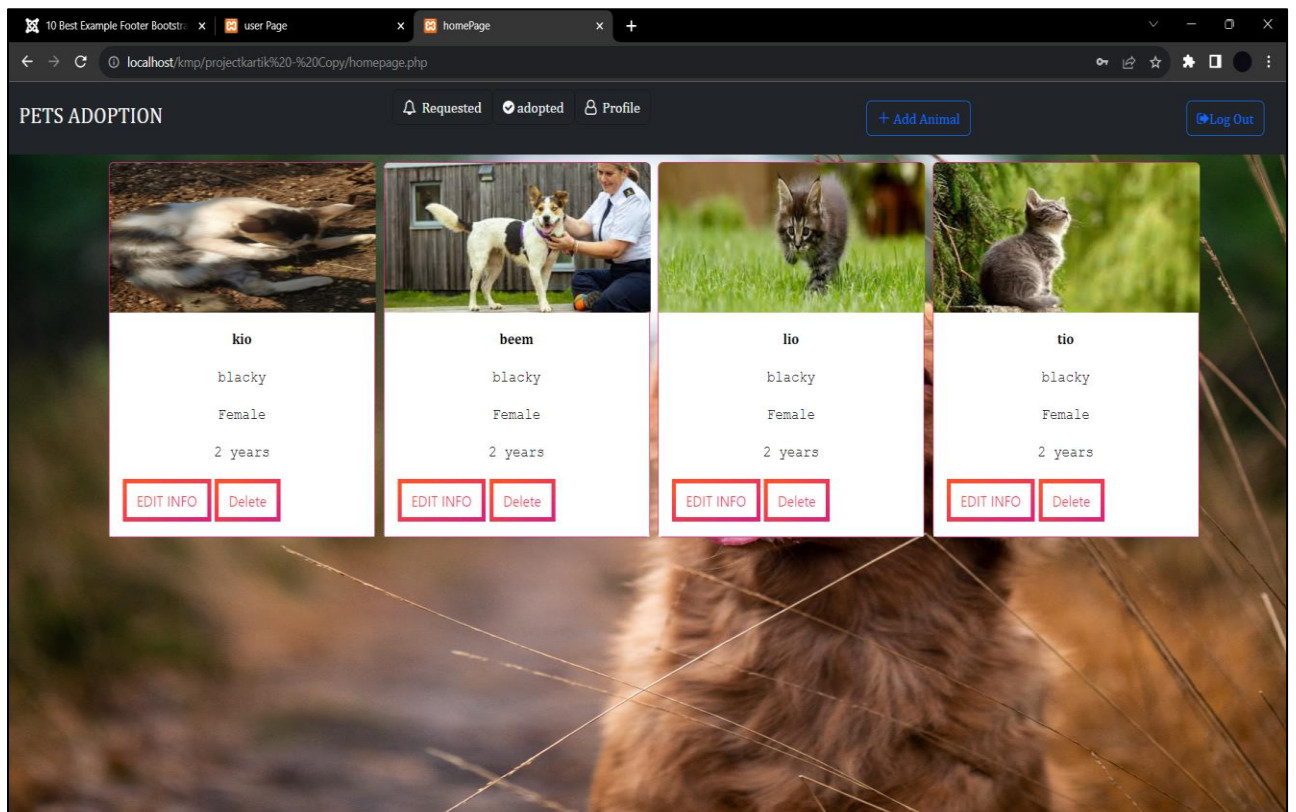
User HomePage



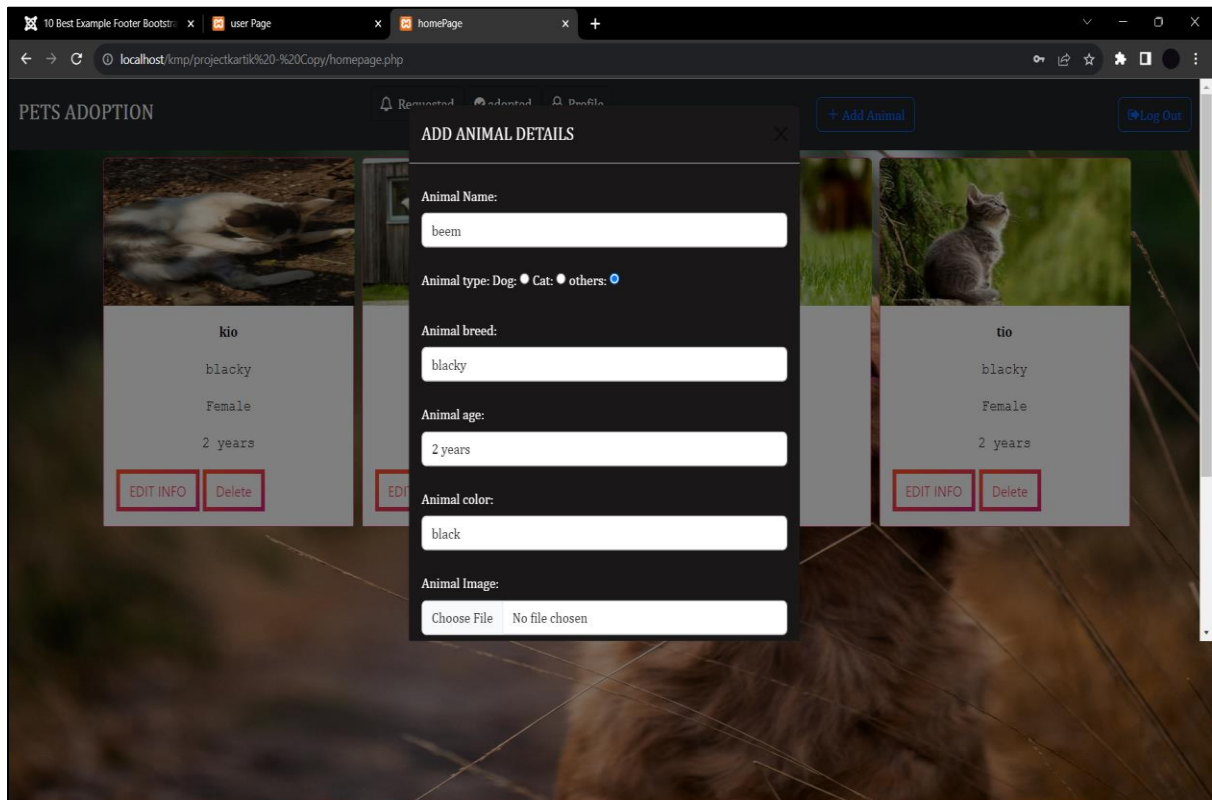
Profile Page



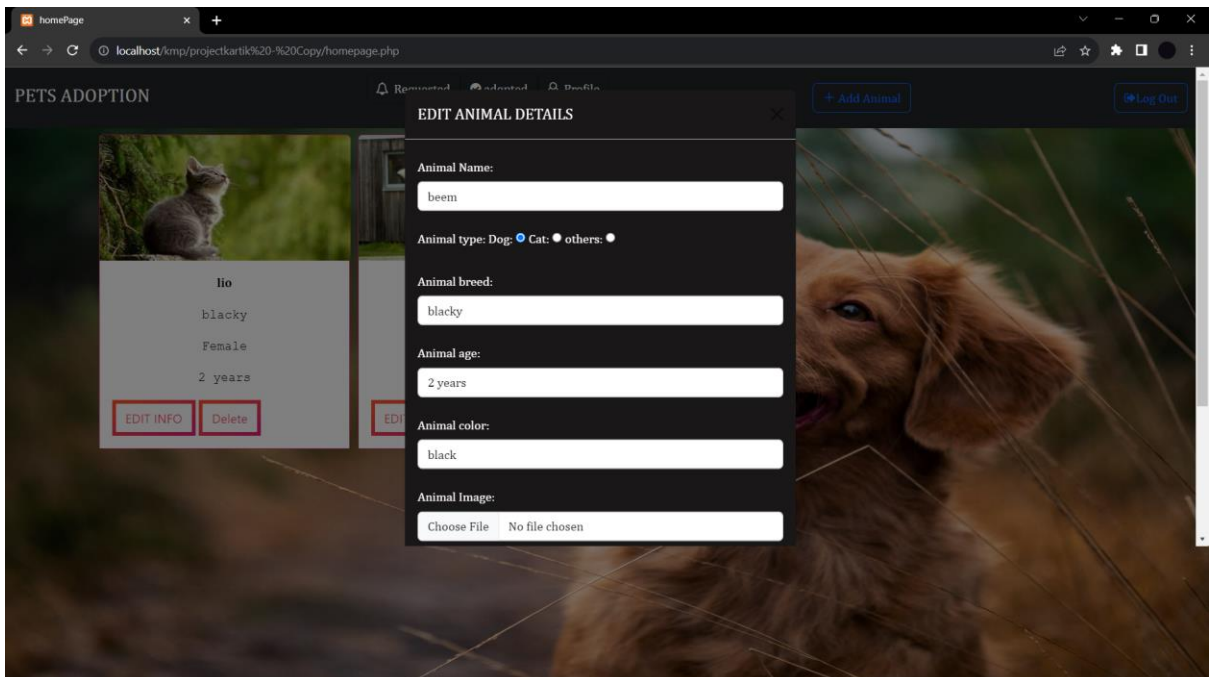
Shelter HomePage



Add Animal



Edit Animal Details



Edit Profile

The screenshot shows a web browser window with the URL `localhost/kmp/projectkartik%20-%20Copy/profile.php`. The page title is "PETS ADOPTION". A modal titled "EDIT DETAILS" is open, displaying a user profile card for "rony sawantwadi" in the background. The modal contains the following fields:

- Name:
- Email:
- phone no:
- Address:

An "update" button is located at the bottom of the modal.

Password Change

The screenshot shows the same web browser window as above. The "EDIT DETAILS" modal is open, but the background profile card is partially obscured. The modal contains the following fields:

- Current Password:
- New Password:
- Confirm Password:

An "update" button is located at the bottom of the modal.

19. CONCLUSION

In conclusion, a well-designed and responsibly managed pet adoption website has the potential to make a significant positive impact on the lives of animals and their future owners. By providing a comprehensive platform that includes detailed pet profiles, user-friendly interfaces, and educational resources, the website can facilitate successful adoptions and contribute to the well-being of pets.

However, it's crucial to acknowledge the inherent limitations and challenges associated with such a platform. Geographical constraints, legal considerations, financial limitations, and the need for accurate information verification are all factors that must be carefully addressed to ensure the website's effectiveness and credibility.

In the face of these challenges, the commitment to responsible pet adoption practices, compliance with regulations, and the implementation of robust security measures are paramount. By fostering a sense of community, encouraging responsible ownership, and working collaboratively with shelters and rescue organizations, the website can become a valuable resource for both prospective pet owners and the animals in need of loving homes.

20.REFERENCES

Following websites have been Referred to create this project report.

- <https://www.w3schools.com/>
- <https://www.php.net/>
- <https://getbootstrap.com/>
- <https://www.mysql.com/>
- <https://www.youtube.com/>
- <https://www.geeksforgeeks.org/>