|  |
| --- |
| Assignment 5 – PHP, MySQL, HTML, CSS, and Bootstrap 5 |

Animal adoption is a noble act of taking care of lost or abandoned animals, such as cats and dogs, who cannot fend for themselves. These animals are usually picked up by animal control authorities and taken to animal shelters where they can receive proper care. To make animal adoption easier and more convenient, a web-based adoption site can be developed that features comprehensive information about the animals available for adoption.

The website should include photos and detailed descriptions of each animal to give potential adopters an idea of their personalities, habits, and preferences. Moreover, the website should offer an online application process for potential adopters, which can save a lot of time and effort. To make the web-based adoption site accessible to everyone, it can be designed to be user-friendly and easy to navigate.

Users of the system:

The system provides two distinct user groups - potential pet adopters and administrators responsible for managing the web application.

The requirements have been delineated as follows:

User:

1. Users should be able to register themselves in the system to view a list of animals waiting to be adopted, along with each animal's records, such as breed, name, estimated age, gender, and medical history.

2. Users should be able to log in/log out to the system.

3. Users should be able to search for available dogs by breed or age to find their perfect pet match.

4. Users should be able to view a list of animals in the shelters and their basic information to make decisions about adoption.

5. Users should be able to send messages or ask questions through the system.

6. Users should be able to request to adopt an animal through the system (fill out the form).

7. Users should be notified of the adoption results by email and confirmation message.

Administrator:

1. The following are the responsibilities of the administrator in charge: adding, editing, updating, and deleting information if necessary.
2. In addition, this administrator should have the ability to determine whether a potential adopter meets the qualifications necessary to raise a pet.
3. They should also be able to send adoption results to users via email through the system.
4. The administrator should be able to confirm whether the user has come to adopt the animal or canceled their adoption request.

**Design**

A diagram of a cell phone

Description automatically generated

**Database Tables**

A screenshot of a computer

Description automatically generated

See SQL file.

CREATE TABLE IF NOT EXISTS Dogtable (

`DogID` INT NOT NULL AUTO\_INCREMENT,

`DogName` VARCHAR(45) NOT NULL,

`DogBreed` VARCHAR(45) NOT NULL,

`DogSex` VARCHAR(8) NOT NULL,

`DogAge` INT NOT NULL,

`DogColor` VARCHAR(15) NOT NULL,

`DogSize` VARCHAR(15) NOT NULL,

`DogDescription` VARCHAR(45) NOT NULL,

`DogMedical` VARCHAR(45) NOT NULL,

PRIMARY KEY (`DogID`));

-- -----------------------------------------------------

-- Table`UserTable`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS UserTable (

`UserID` INT NOT NULL AUTO\_INCREMENT,

`Username` VARCHAR(15) NOT NULL,

`FirstName` VARCHAR(45) NOT NULL,

`LastName` VARCHAR(45) NOT NULL,

`Password` VARCHAR(15) NOT NULL,

`UserType` VARCHAR(20) NOT NULL,

`AnnualIncome` DOUBLE NULL DEFAULT NULL,

`Email` VARCHAR(30) NOT NULL,

`Address` VARCHAR(45) NOT NULL,

`NumberOfPetsAtHome` INT NULL DEFAULT NULL,

PRIMARY KEY (`UserID`));

-- -----------------------------------------------------

-- Table `History`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS History (

`HistoryID` INT NOT NULL AUTO\_INCREMENT,

`AdoptionDate` DATE NOT NULL,

`AdoptionStatus` VARCHAR(15) NOT NULL,

`HistoryNote` VARCHAR(45) NOT NULL,

`Dogtable\_DogID` INT NOT NULL,

`UserTable\_UserID` INT NOT NULL,

PRIMARY KEY (`HistoryID`, `Dogtable\_DogID`, `UserTable\_UserID`),

INDEX `fk\_History\_Dogtable1\_idx` (`Dogtable\_DogID` ASC) ,

INDEX `fk\_History\_UserTable1\_idx` (`UserTable\_UserID` ASC) ,

CONSTRAINT `fk\_History\_Dogtable1`

FOREIGN KEY (`Dogtable\_DogID`)

REFERENCES `myronovy`.`Dogtable` (`DogID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_History\_UserTable1`

FOREIGN KEY (`UserTable\_UserID`)

REFERENCES `myronovy`.`UserTable` (`UserID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION);

-- -----------------------------------------------------

-- Table MessageTabel`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS MessageTabel (

`MessageID` INT NOT NULL AUTO\_INCREMENT,

`UserID` INT NOT NULL,

`UserName` VARCHAR(45) NOT NULL,

`MessageDate` DATE NOT NULL,

`MessageEmail` VARCHAR(45) NOT NULL,

`MessageText` VARCHAR(100) NOT NULL,

`UserTable\_UserID` INT NOT NULL,

PRIMARY KEY (`MessageID`),

INDEX `fk\_MessageTabel\_UserTable1\_idx` (`UserTable\_UserID` ASC) ,

CONSTRAINT `fk\_MessageTabel\_UserTable1`

FOREIGN KEY (`UserTable\_UserID`)

REFERENCES `myronovy`.`UserTable` (`UserID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION);

**GUI design and expectations**

|  |  |  |
| --- | --- | --- |
| N | Requirements | Points |
| 1 | The home page should include:   1. Login, Register, About, and Contact hyperlinks. 2. Picture/Pictures of dogs. 3. Welcome message. 4. Header and footer. Use Bootstrap. 5. The footer should include student names, class number, and year.   Note: show all dogs for adoption only to registered users. | 10 |
| 2 | The Login page should include:   1. Form with Login input boxes: username and password. 2. Button Submit. 3. Link to Register web page. 4. Header and Footer. | 10 |
| 3 | The Register web page should include:   1. Header and footer. 2. HTML Form: 3. `Username` 4. `FirstName` 5. `LastName` 6. `Password` 7. `UserType` - user or administrator 8. `AnnualIncome` 9. `Email` 10. `Address` 11. `NumberOfPetsAtHome`      1. Button Submit. 2. Button Reset.   Note: After a user completes the registration process, please direct the current user to the Login page. | 20 |
| 4 | The Dogslist web page should include:   1. Header and footer. 2. Name of the logged-in user. 3. List of ALL available dogs: picture, name, and breed. 4. Search support: Users should be able to search for available dogs by breed or age to find their perfect pet match. 5. The user should be able to view a detailed description of a selected dog in a new window. | 40 |
| 5 | The Request for adoption Form should include:   1. Header and footer. 2. HTML Form: 3. `AdoptionStatus` - pending/adopted 4. `HistoryNote` 5. Button Submit. 6. Button Reset. | 30 |
| 6 | The Communication/Message page should include:   1. Header and footer. 2. HTML “Submit a new Message “ Form: 3. Username – read from the PHP Session 4. Email – can be different. 5. Message – use text area HTML element. 6. List of all Messages for the current User. 7. Button Submit. 8. Button Reset. | 40 |
| 7 | The administrator page should include:   1. Header and footer. 2. Hyperlinks to “Add a Dog”. 3. Manage the list of Dogs – update and delete. 4. Communication/Message page. | 30 |
| 8 | The “Add a new Dog” page should include:   1. Header and footer. 2. HTML form to add a new dog to the table. 3. Button Submit. 4. Button Reset. 5. Link to the Communication/Message page. | 30 |
| 9 | The “Manage the list of Dogs”, update and delete, should include:   1. Header and footer. 2. The table with all available dogs: DogID, Name, and picture. 3. HTML form to delete dog from the database table. 4. HTML form to update selected dog information from the database table. 5. Link to the Message page. 6. Button Submit. 7. Button Reset. 8. Link to the Communication/Message page. | 40 |