



**Bilkent University**

**Department of Computer Engineering**

---

# CS353 Term Final Report

**DatAnimal**

**Assigned Teacher Assistant**

Duygu Durmuş

Group 34

**Team Members**

Asım Güneş Üstüinalp

Radman Lotfiazar

Turan Mert Duran

Berdan Akyürek

<b>Introduction</b>	<b>3</b>
<b>Final ER Diagram</b>	<b>4</b>
<b>Final List of Tables</b>	<b>4</b>
<b>Implementation Details</b>	<b>8</b>
<b>Advanced Database Features</b>	<b>10</b>
<b>User's Manual</b>	<b>12</b>
Register	12
Login	15
Coordinator	16
Facility Worker	24
Keeper	27
Veterinarian	33
Visitor	34
Visitor Home page	34
Donate	35
Attend Group Tour	37
Wallet	40
Complaint	41

# 1. Introduction

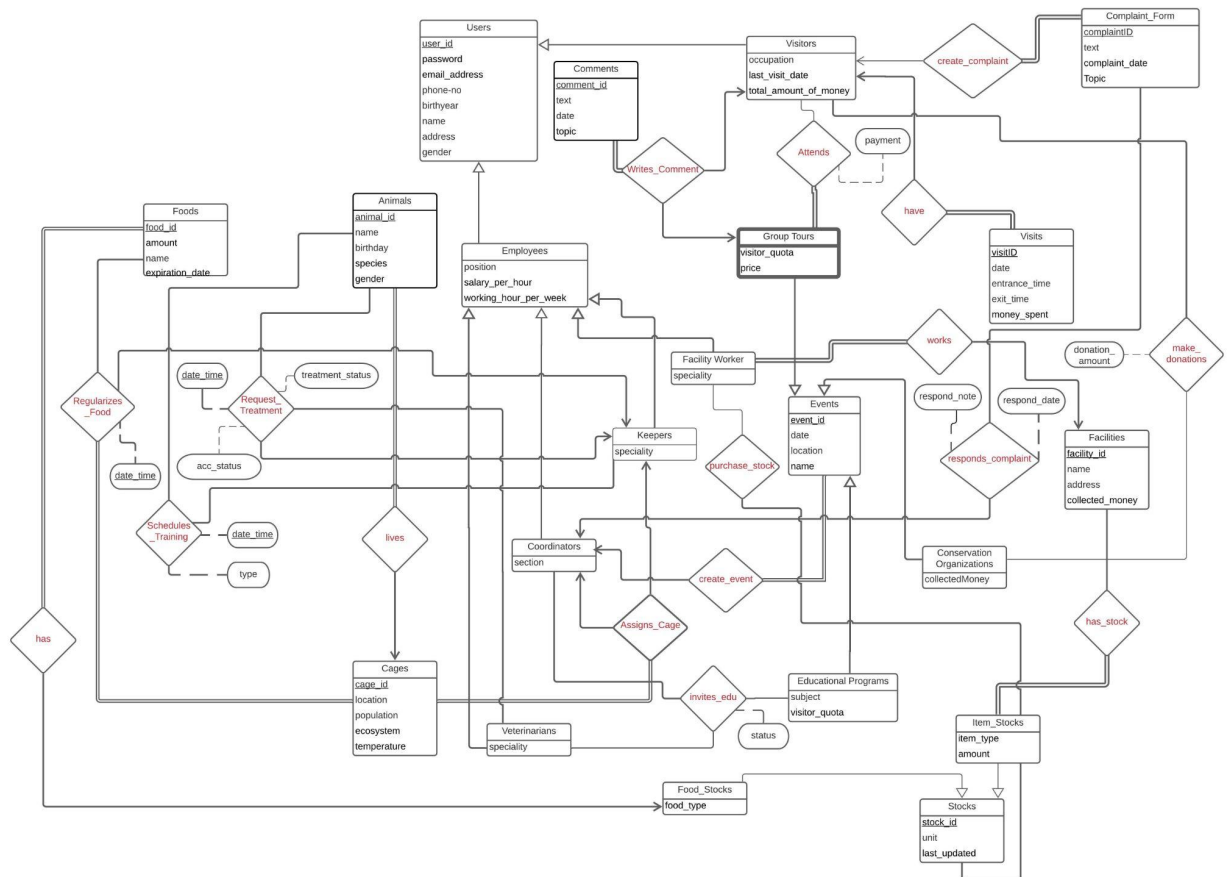
DatAnimal is a database system that was created to make the management of a Zoo easier. It stores the necessary information in a Mysql database. The system can be used by 5 different types of users. Those are; visitors, coordinators, keepers, veterinarians and facility workers. Depending on the user's type the operations that the user can perform changes.

For instance, the visitors can attend group tours, write comments and create complaints. The keepers can regulate the foods for the cages, request treatment for the animals in their cages and schedule training for the mentioned animals. Coordinators can assign cages to keepers, create events, invite veterinarians to educational programs. Veterinarians can respond to the treatment requests made by the keepers and can respond to the educational program invitations made by the coordinators. Facility workers can purchase stock for the facilities of the Zoo( restaurants, toilets etc.).

We implemented all the functionalities that were present on the ER Diagram. Other than the users, the system keeps track of many different things such as the information about the animals and the cages or information about the complaints and the corresponding response.

As a group, we regularly held meetings. We drew the ER diagram together on a Zoom call. For the design report we divided the sections equally amongst ourselves. For the implementation we divide the project into parts. Each group member did their part and then we merged them together. We used Github extensively in this process. The way we divided the implementation part was like this: Radman did the Visitor page and all the operations that a visitor can perform, Berdan did the Coordinator and the Facility worker pages and all their functionalities, Turan did the Login and Register pages as well as the Veterinarian page and all its functionalities. Gunes did the Keeper page and all its functionalities. For the final report, every group member wrote the user manual section of the pages they created. The rest of the report has been divided equally. Our code can be found at: <https://github.com/DatAnimal-Bilkent/DatAnimal-Bilkent.github.io>

## 2. Final ER Diagram



In case of any confusion, a bigger version of ER diagram can be found on the project website (<https://datanimal-bilkent.github.io/>).

## 3. Final List of Tables

Users( user\_id, password, email, phone\_no, birth\_year, name, address, gender)

Employees( user\_id, position, salary\_per\_hour, working\_hour\_per\_week)

foreign key: (user\_id) references Users(user\_id)

Visitors(user\_id, occupation, last\_visit\_date, total\_amount\_of\_money)  
foreign key: (user\_id) references Users(user\_id)

Facilities(facility\_id, name, address, collected\_money)

Facility\_Worker(user\_id, speciality, facility\_id )  
foreign key: (user\_id) references Employees(user\_id)  
foreign key: (facility\_id) references Facilities(facility\_id)

Keepers(user\_id, speciality)  
foreign key: (user\_id) references Employees(user\_id)

Coordinators(user\_id, section)  
foreign key: (user\_id) references Employees(user\_id)

Veterinarians(user\_id, speciality)  
foreign key: (user\_id) references Employees(user\_id)

Cages(cage\_id, location, population, ecosystem, temperature)

Animals(animal\_id, name, birthday, cage\_id, species, gender)  
foreign key: (cage\_id) references Cages(cage\_id)

Stocks(stock\_id, unit, last\_update)

Food\_Stocks(stock\_id, food\_type)  
foreign key: (stock\_id) references Stocks(stock\_id)

Foods(food\_id, name, expiration\_date, stock\_id, amount)  
foreign key: (stock\_id) references Food\_Stocks(stock\_id)

Item\_Stocks(stock\_id, item\_type, amount)

foreign key: (stock\_id) references Stocks(stock\_id)

Events(event\_id, date, location, user\_id, name)

foreign key: (user\_id) references Coordinators(user\_id)

Group\_Tours(event\_id, visitor\_quota, price)

foreign key: (event\_id) references Events(event\_id)

Educational\_Program(event\_id, visitor\_quota, subject)

foreign key: (event\_id) references Events(event\_id)

Conservation\_Organizations(event\_id, collectedMoney)

foreign key: (event\_id) references Events(event\_id)

Comments(comment\_id, topic, text, date)

Complaint\_Form(complaint\_id, topic, text, complaint\_date, user\_id)

foreign key: (user\_id) references Visitors(user\_id)

Respond\_complaint(complaint\_id, user\_id, respond\_note, respond\_date)

foreign key: (complaint\_id) references Complaint\_Form(complaint\_id)

foreign key: (user\_id) references Coordinators(user\_id)

Visit(visit\_id, date, entrance\_time, exit\_time, money\_spent, user\_id)

foreign key: (user\_id) references Visitors(user\_id)

Schedules\_Training(animal\_id, user\_id, date\_time, type)

foreign key: (animal\_id) references Animals(animal\_id)

foreign key: (user\_id) references Keepers(user\_id)

Regularizes\_Food(animal\_id, user\_id, cage\_id, food\_id, date\_time)

foreign key: (food\_id) references Foods(food\_id)

foreign key: (user\_id) references Keepers(user\_id)

foreign key: (cage\_id) references Cages(cage\_id)

Request\_Treatment(animal\_id, keeper\_user\_id, vet\_user\_id, date\_time, cage\_id, treatment\_status, acc\_status)

foreign key: (keeper\_user\_id) references Keepers(user\_id)

foreign key: (vet\_user\_id) references Veterinarians(user\_id)

foreign key: (cage\_id) references Cages(cage\_id)

Writes\_Comment(comment\_id, event\_id, user\_id)

foreign key: (comment\_id) references Comments(comment\_id)

foreign key: (event\_id) references Group\_Tours(event\_id)

foreign key: (user\_id) references Visitors(user\_id)

Attends(user\_id, event\_id, payment)

foreign key: (event\_id) references Group\_Tours(event\_id)

foreign key: (user\_id) references Visitors(user\_id)

Assigns\_Cage(coordinator\_user\_id, keeper\_user\_id, cage\_id)

foreign key: (coordinator\_user\_id) references Coordinators(user\_id)

foreign key: (keeper\_user\_id) references Keepers(user\_id)

foreign key: (cage\_id) references Cages(cage\_id)

Make\_Donation(event\_id, user\_id, donation\_amount)

foreign key: (event\_id) references Conservation\_Organizations(event\_id)

foreign key: (user\_id) references Visitors(user\_id)

Purchase\_Stock(user\_id, stock\_id)

foreign key: (user\_id) references Facility\_Worker(user\_id)

foreign key: (stock\_id) references Stocks(stock\_id)

Invites\_Edu(vet\_user\_id, coor\_user\_id, event\_id)

foreign key: (coor\_user\_id) references Coordinators(user\_id)

foreign key: (vet\_user\_id) references Veterinarians(user\_id)

foreign key: (event\_id) references Educational\_Program(event\_id)

Has\_Stock(stock\_id, facility\_id)

foreign key: (stock\_id) references Item\_Stocks(stock\_id)

foreign key: (facility\_id) references Facilities(facility\_id)

## 4. Implementation Details

As we stated in Proposal we implemented our project by getting help from MySQL for database and PHP 8.0.5 for front end. By MySQL we created our table and with its query we were able to insert and update our table throughout the Application.

There is a class in PHP, mysqli which helps us to connect and update the database. For instance, the function `mysqli_connect()` by taking the URL, password and username of the database as parameters allows you to connect to the database. Also for updating MySQL, function `mysqli_query(connection, query)` by taking a connection variable which includes details about your connections and query let you access the database and make changes stated in the query.

Moreover, since variables in PHP do not need to have type such as integer, character or string, we are able to store our query as a string into a variable and use them in the `mysqli_query` function. Also, we are able to store connection information which is returned by the `mysqli_connect()` in a variable and use it as a parameter of the `mysqli_query()` function from Mysqli class. Use of this class throughout the PHP helps us to access, update, delete the data in the database in the fastest way.

Nevertheless, finding out the syntax error or logic error in the query of MySQL was the most difficult part of the project because we did not face any error throughout the webpage or terminal. Therefore, sometimes we had to spend lots of time finding out a



problem due to a query. However, dijkstra provides errors about the query and sometimes this tool helps us find problems in the more convenient way. While PHP helps us to indicate the data in the database, for showing the data in a better way we also use HTML and JavaScript codes. These two programming languages by providing different functionality such as header, table, button and link improve our User Interface.

Lastly, in order to implement email to visitors function, we used a library called PHPMailer. Although PHP already has its built in mail function, it was not possible to use it in our case because that function required us to run our own mail server. Since this was not sufficient for us, we used PHPMailer that makes email sending faster using outside servers like Gmail. Also it was easy to install using the Composer tool and easy to learn thanks to its simplicity.

## 5. Advanced Database Features

-We used secondary indices for all the user tables such as Keepers, Veterinarians, Coordinators, Facility\_Workers, Visitors. Beside their user\_id, we gave them a secondary index with their name. This way, we didn't require access to the Users table to learn a user's name every time we required a user's name.

-We used the MAX feature of SQL.

```
"SELECT MAX(event_id)+1 as tmp from Events";
```

We use this query when creating a new event. To assign an id to the new event, we first find the max event\_id in the table and give the new event that max id + 1.

-We used the LIKE feature of SQL.

```
SELECT DISTINCT event_id, name, location, date, CollectedMoney  
FROM Events NATURAL JOIN Conservation_Organizations  
WHERE name LIKE '%$organ%';
```

The screenshot shows a web browser at localhost:8000/Visitors/Donate.php? with a dark theme. At the top, there are links for 'Logout' and 'Home Page'. The main heading is 'We Will be Happy if You Donate Our Organization'. Below this, it says 'Your Credit: 1360\$'. A section titled 'Your Past Donations' contains a table with 4 columns: Event Name, location, Date of Organization, and Total of your Donation. The table has 3 rows of data. Below this is a section titled 'Organizations For Your Donation' which includes a search bar with the placeholder 'Organization org' and a 'Search' button. At the bottom of this section is another table with 5 columns: Event Name, location, Date of Organization, Total Donated Money, and Donate. This table has 2 rows of data. A 'Back' link is at the very bottom.

Event Name	location	Date of Organization	Total of your Donation
org1	entrance	2021-05-05	100 \$
asd	entrance	2021-05-12	12 \$
asd	qwe	2021-05-12	200 \$

Event Name	location	Date of Organization	Total Donated Money	Donate
org1	entrance	2021-05-05	100 \$	Donate
org1	entrance	2021-05-05	0 \$	Donate

We used the LIKE feature to reduce errors due to typos made by the user. The users can find conservation organizations even if they made a minor mistake while typing its name.

Above, you can see that the user typed “org” but could see the events named “org1”.

-We used the delete query of SQL.

```
DELETE FROM Assigns_Cage WHERE cage_id=$cd
```

We used this query to unassign a cage. The cage assignments are kept in the Assigns\_Cage table, with this statement we unassign the cage by deleting the associated row.

-We used the update query of SQL.

```
UPDATE Facilities SET collected_money = $remainingMoney WHERE  
facility_id = $facid;
```

```
UPDATE Group_Tours SET visitor_quota = '$quota' WHERE event_id  
= '$org_id';
```

We used the update feature several times. Here are two examples.

For the above one, it is used to update the Facilities table when the facility worker makes a purchase. The collected money of the facility worker’s Facility is decreased.

For the below one, it is used to update the Group\_Tours table when a visitor registers to a Group Tour. The quota of the Group Tour is decreased when a visitor registers to a Group Tour. The outputs of these operations can be found in their respective user manual entries.

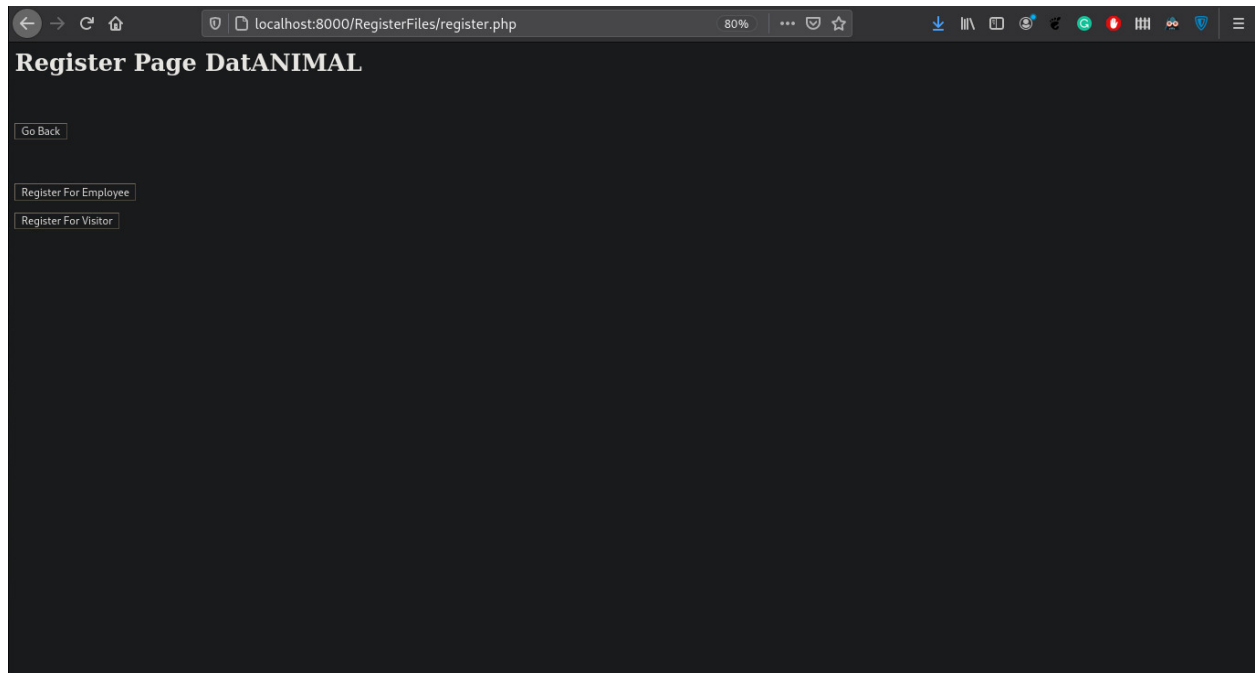
-We used the IN feature of SQL.

```
SELECT * FROM Cages WHERE cage_id NOT IN (SELECT cage_id FROM  
Assigns_Cage);
```

We used this query to display the unassigned cages in our database. This is used in the assign cage page of the coordinator. The outputs can be seen in the page’s user manual entry.

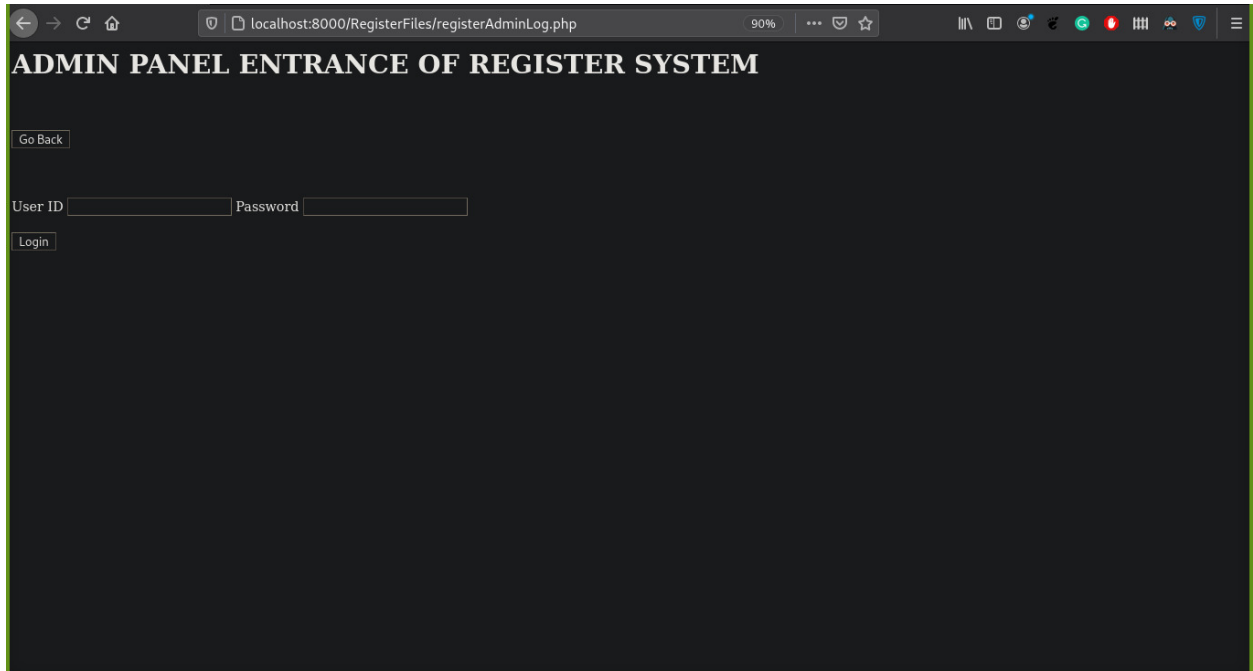
## 6. User's Manual

### 6.1. Register



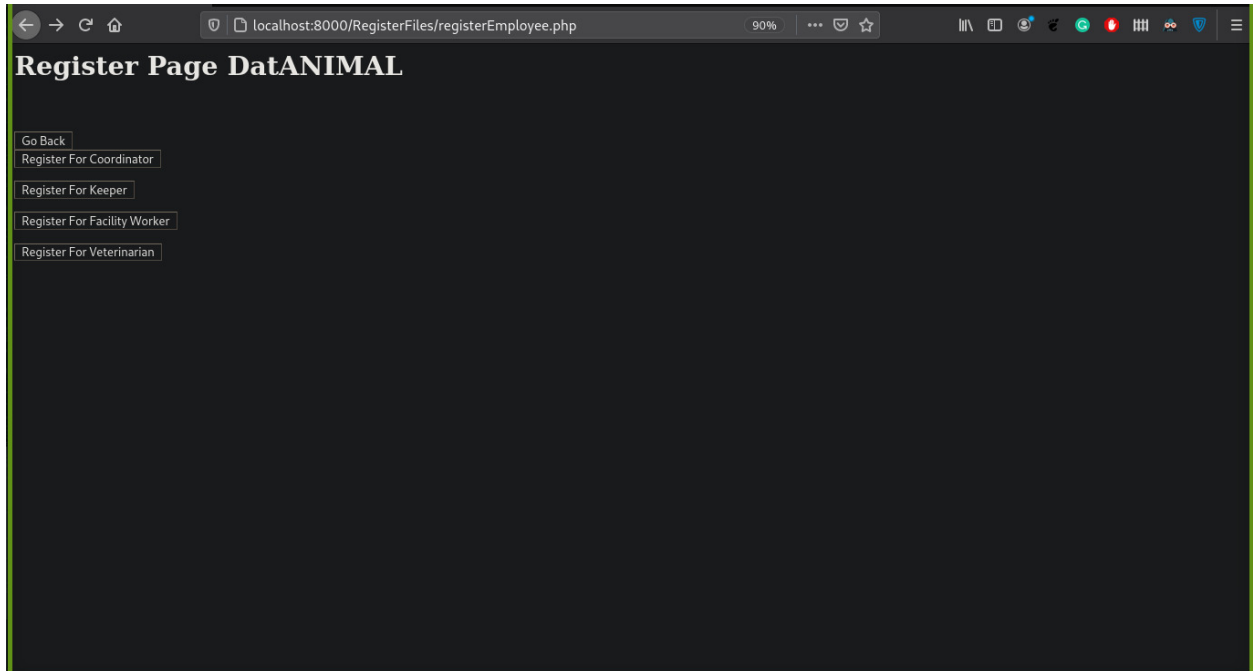
Picture 6.1.1

When the user presses the “Register” button, this page appears. The user can press either the “Register for Employee” or the “Register for Visitor”. There is also the “Go Back” button that allows the user to return to the login page.



Picture 6.1.2

When the user presses the “Register for Employee” button on the register page, they are directed to this page. The user needs to provide an admin id and an admin password in order to progress to the register for employee page. The reason for this is to prevent non-employees from registering as employees. An admin needs to be present when an employee account will be created.



Picture 6.1.3

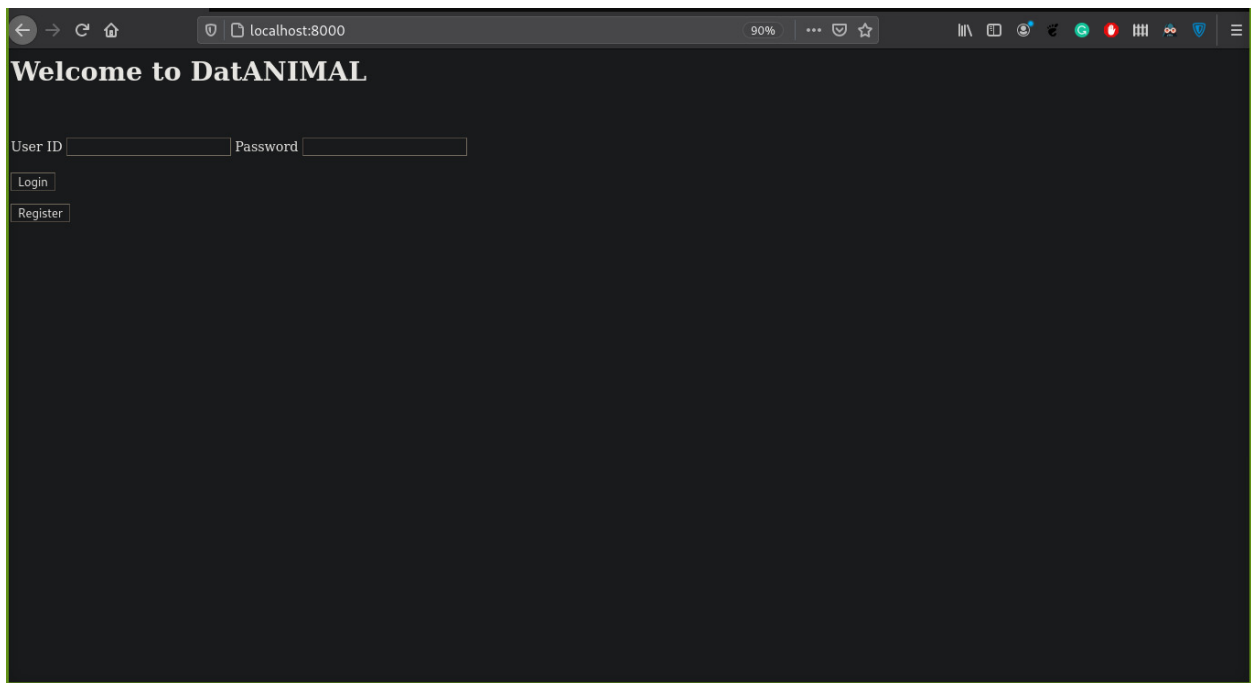
This is the register page for employees. The user can select which kind of employee they want to register as. When they click a register button they will be redirected to the register page of the associated employee type.

A screenshot of a web browser displaying the 'KEEPER REGISTER SYSTEM'. The browser's address bar shows the URL 'localhost:8000/RegisterFiles/registerKeeper.php'. The page has a dark background with the title 'KEEPER REGISTER SYSTEM' at the top. Below the title, there is a 'Go Back' button followed by a series of input fields: 'User ID' (with a dropdown arrow), 'Password', 'Email', 'PhoneNo' (with a dropdown arrow), 'Birthdate' (with a date picker showing 'mm / dd / yyyy'), 'Name', 'Address', 'Gender', and 'Speciality'. At the bottom of these fields is a 'Submit' button.

Picture 6.1.4

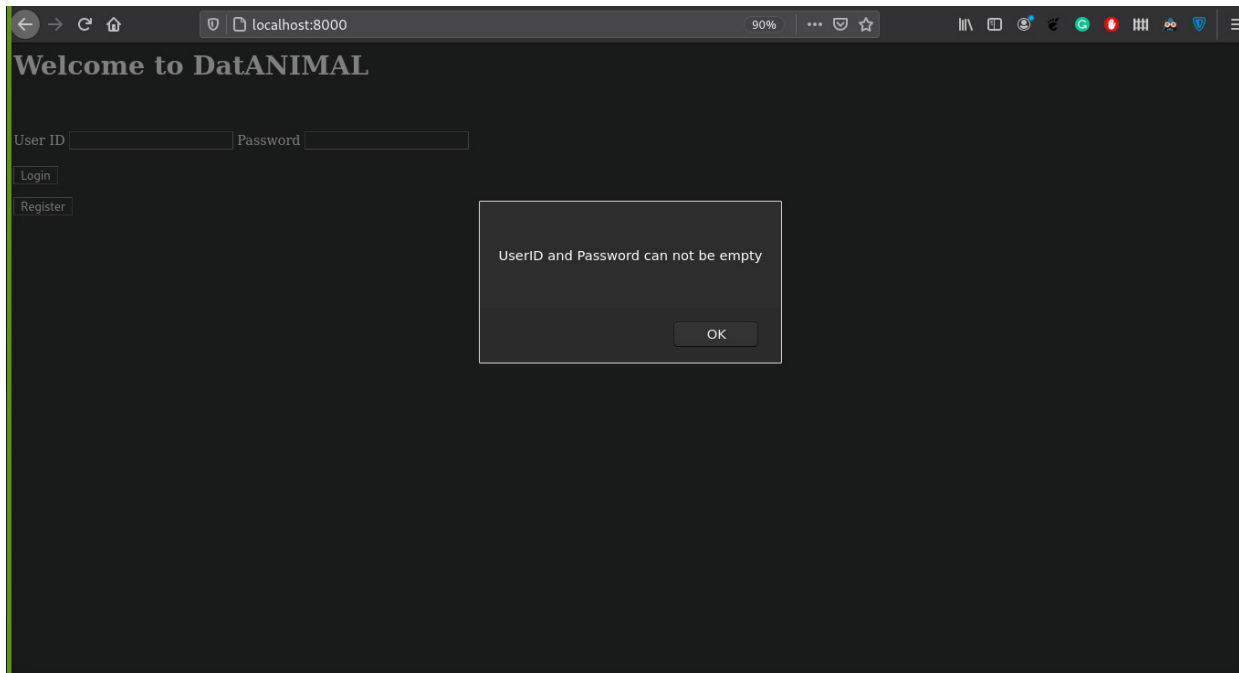
This is the register page for keepers as an example. Every other type of register page is similar to that one, so the manual will only talk about this one as an example. The user needs to fill all the necessary fields before pressing “Submit”. If not, an alert message will be shown. When they fill all the fields and press “Submit”, a new account will be created with the provided values.

## 6.2. Login



Picture 6.2.1

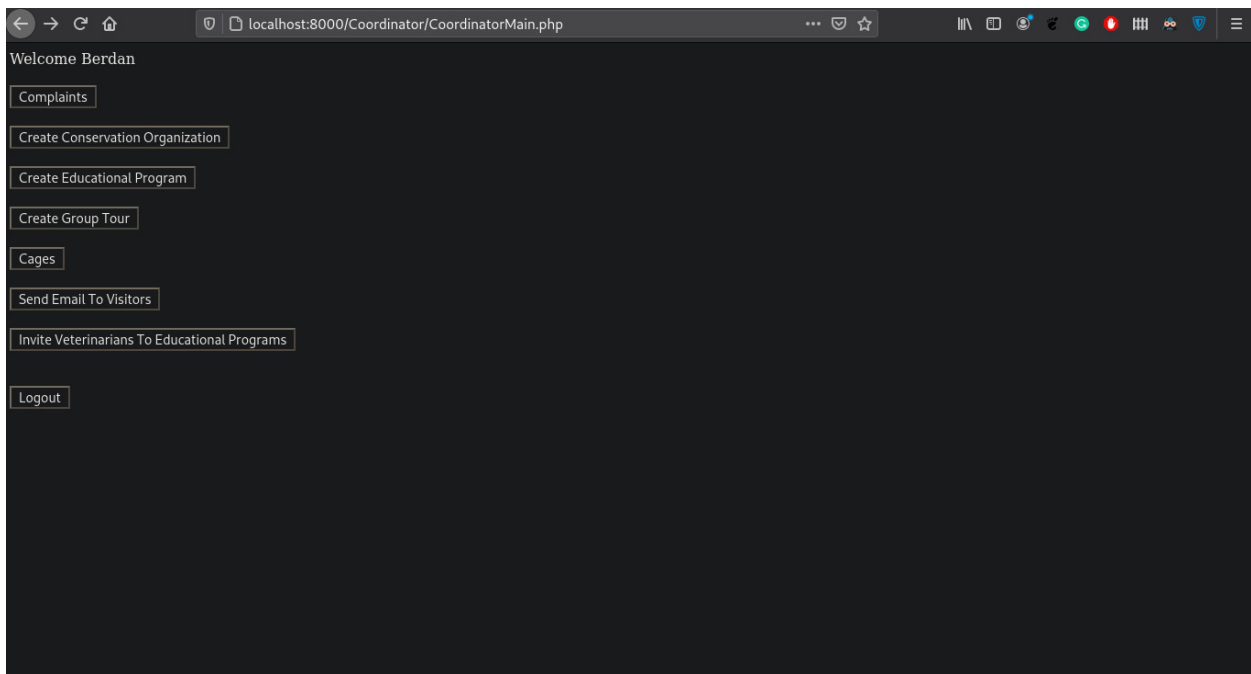
This is the login page of our system. The system opens on this page. The user can enter a user id and a password to the specified fields. When they press “login” if the required fields are not empty, the login result will be shown. If any field is empty, an alert pop-up will be shown. Otherwise, if the userID and password combination is legitimate, the user will successfully login to his/her account. If the combo is wrong, the login attempt will be unsuccessful. If the user presses the “Register” button, the register page will be shown.



Picture 6.2.2

If any field is empty when the login button is pressed, this alert will be shown.

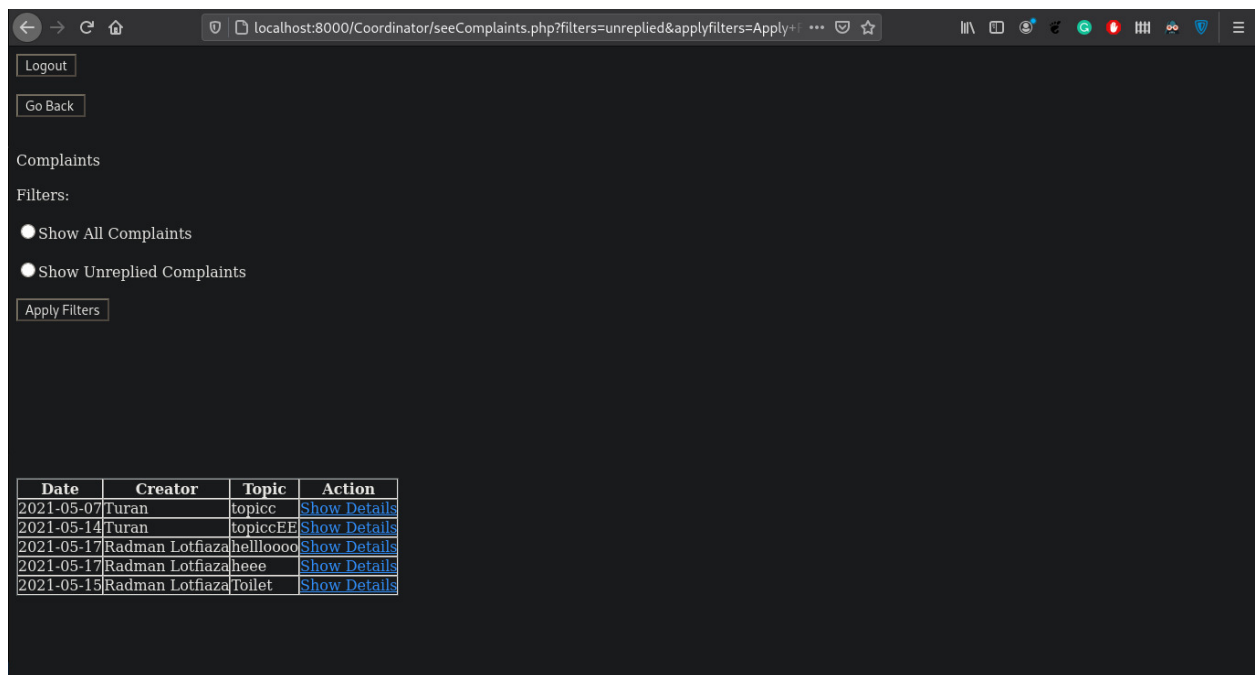
## 6.3. Coordinator



Picture 6.3.1



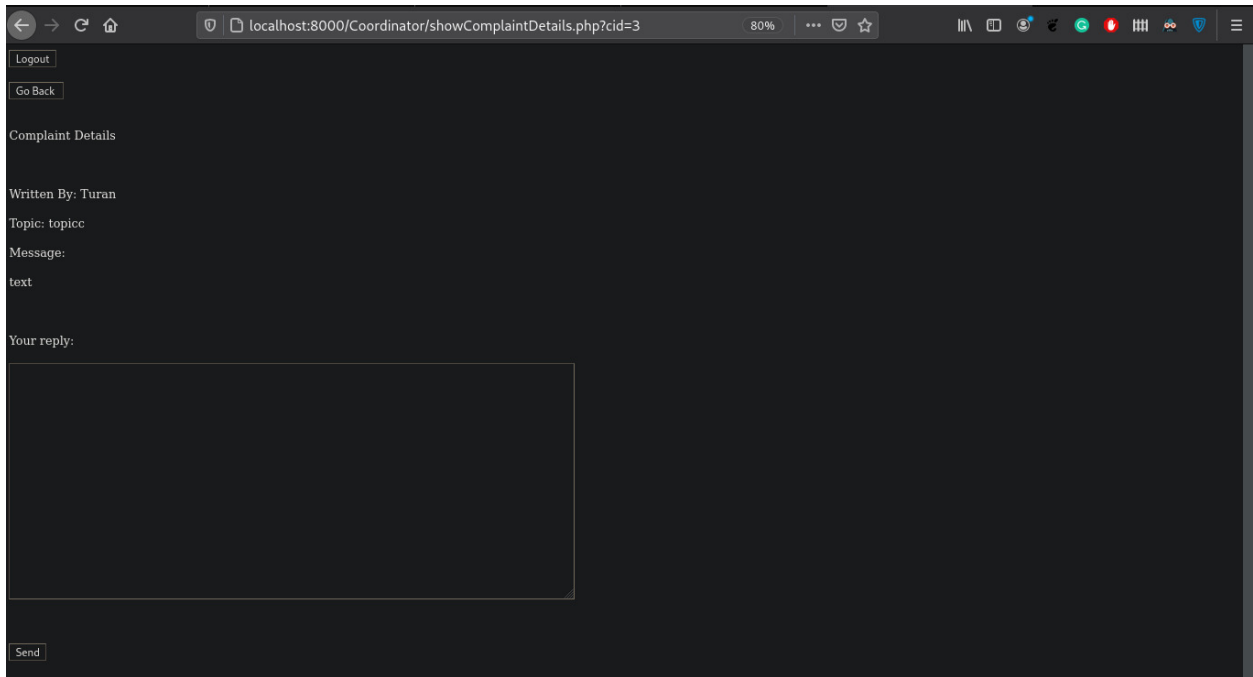
If you log in successfully as a Coordinator, you will see the screen shown in Picture 6.3.1. This is the main screen for Coordinator. You can choose to check complaints, create an event including conservation organizations, educational programs or group tours, see and manage cages, send emails to all visitors that are registered to a group tour, Invite veterinarians to educational programs or Logout using this screen.



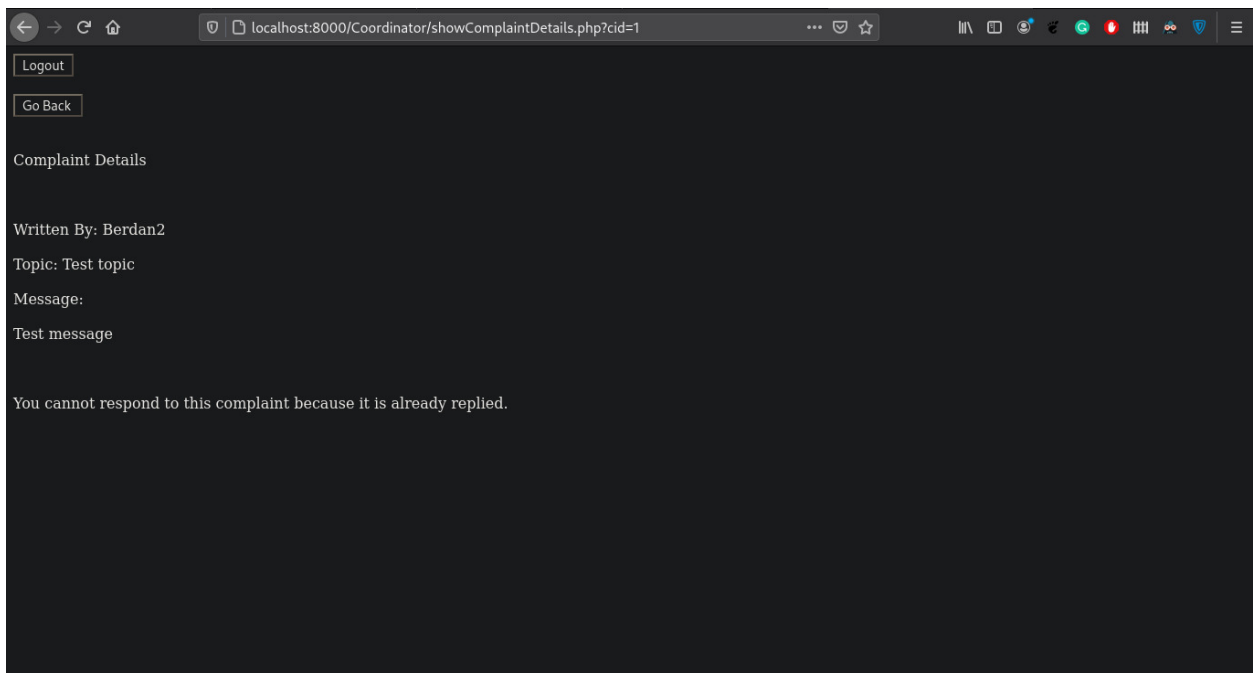
Picture 6.3.2

## **Handle Complaints**

In order to handle complaints, click on the Complaints button on the main screen. If you click on the Complaints button, you will see a list of the complaints sent by visitors as well as some brief information about them. You can choose to list them without any filters or listing only unreplied complaints. To do this, select the filter option that you want to apply and click the Apply Filters button. If you want to get more details about a complaint, click on the related Show Details link.



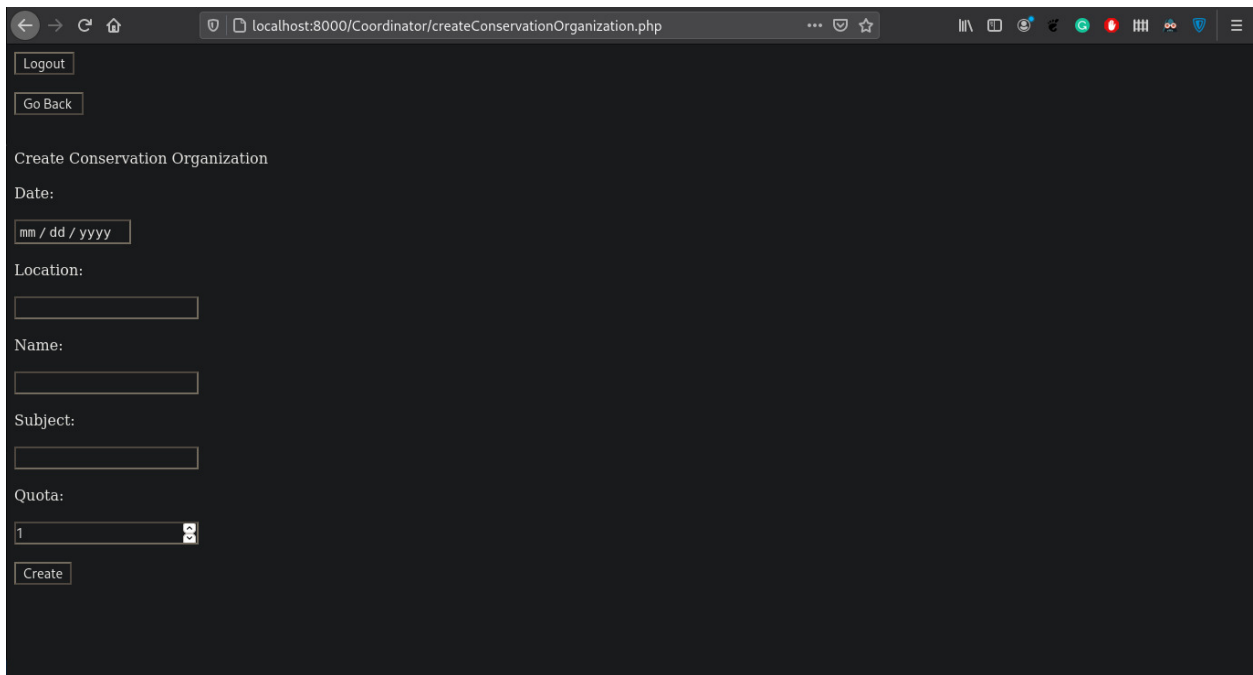
Picture 6.3.3



Picture 6.3.4

After clicking a Show Details Link, you will be redirected to a page that displays all the information about the selected complaint as shown in Picture 6.3.3 and Picture 6.3.4. A single complaint can be replied to only once. If the complaint is not replied to by you or other coordinators, you will see a text field where you can write your reply as shown in Picture 6.3.3. After writing your message, click on the submit button and your reply will be sent to the creator. If the complaint is already replied, you will see a warning that you cannot reply to this complaint below the information of the complaint as shown in Figure 6.3.4.

## **Creating Events**



The screenshot shows a web browser window with the address bar displaying 'localhost:8000/Coordinator/createConservationOrganization.php'. The page has a dark background. At the top left, there are two buttons: 'Logout' and 'Go Back'. Below these, the title 'Create Conservation Organization' is displayed. The form contains the following fields and labels: 'Date:' with a date picker showing 'mm / dd / yyyy'; 'Location:' with a text input field; 'Name:' with a text input field; 'Subject:' with a text input field; and 'Quota:' with a text input field containing the number '1' and a spinner icon. At the bottom left of the form is a 'Create' button.

Picture 6.3.5

A screenshot of a web browser window displaying a form titled "Create Educational Program". The browser's address bar shows the URL "localhost:8000/Coordinator/createEducational.php". At the top left of the page are two buttons: "Logout" and "Go Back". The form contains the following fields: "Date:" with a date picker showing "mm / dd / yyyy"; "Location:" with a text input field; "Name:" with a text input field; "Quota:" with a numeric input field containing the value "1"; and "Subject:" with a text input field. At the bottom left of the form is a "Create" button.

Picture 6.3.6

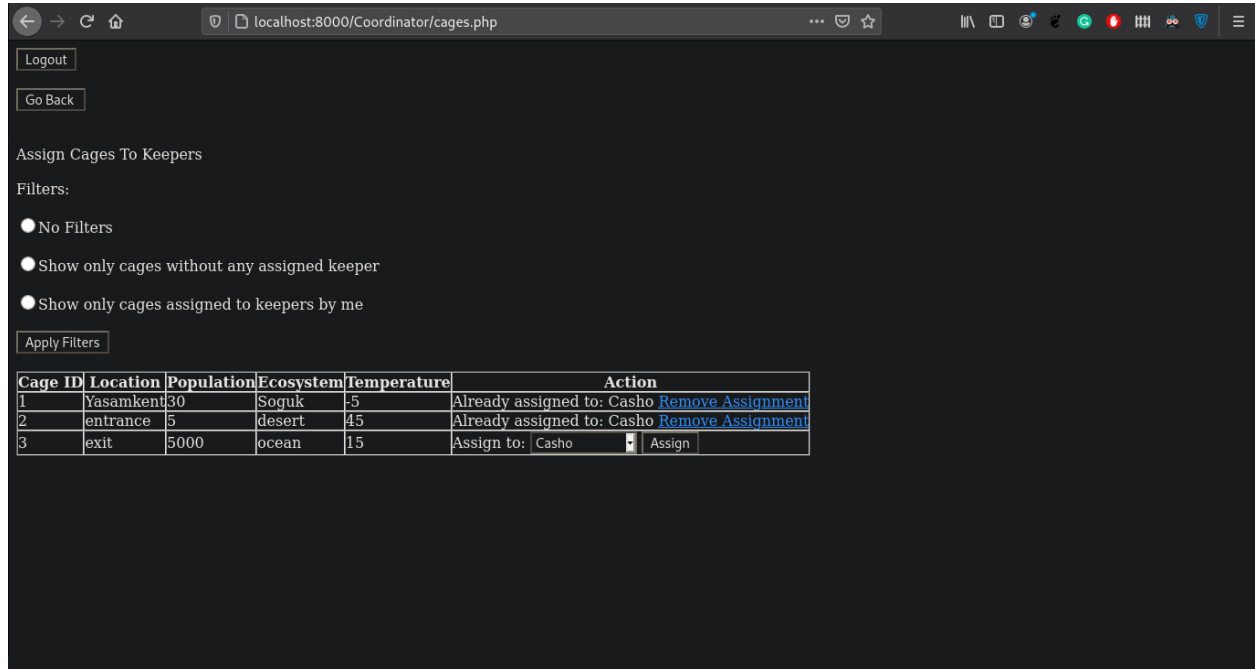
A screenshot of a web browser window displaying a form titled "Create Group Tour". The browser's address bar shows the URL "localhost:8000/Coordinator/createGroupTour.php". At the top left of the page are two buttons: "Logout" and "Go Back". The form contains the following fields: "Date:" with a date picker showing "mm / dd / yyyy"; "Location:" with a text input field; "Name:" with a text input field; "Quota:" with a numeric input field containing the value "1"; and "Price:" with a numeric input field containing the value "1". At the bottom left of the form is a "Create" button.

Picture 6.3.7

In order to create a new event, click the create button with the type of the event you want to create on the main screen. Later you will be redirected to a screen where you

can enter the details of the event as shown in Picture 6.3.5, Picture 6.3.6 and Picture 6.3.7 depending on the type of event you want to create. Provide all necessary information and click the submit button below. If you fill all the fields, you will successfully create an event and will be redirected to the main screen.

## Cage Assignments



Picture 6.3.8

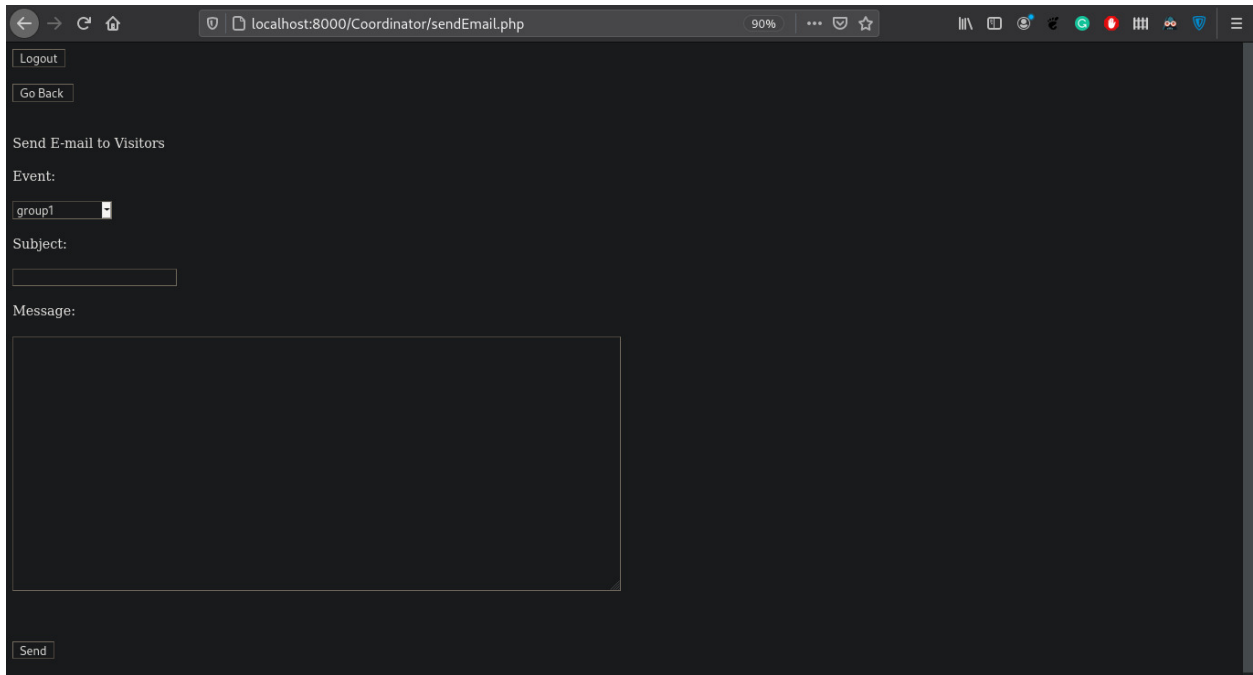
If you want to handle cages and their assigned keepers, click on the Cages button on the main screen. This button will redirect you to the page shown in Picture 6.3.8. Using this page, you can find some information about cages and assigned keepers on them, if any. You can choose to see all cages, only unassigned cages and only assigned by you using radio buttons and Apply Filters button on top.

You can assign a keeper to a cage using this screen. To do this, select a keeper from the dropdown menu next to the cage you want to assign and click assign button. This will assign the selected keeper to the selected cage.

Also, you can cancel the assignment of a keeper to a cage using the Remove Assignment link next to the information of the cage you want to cancel its assignment.

You can leave this screen any time using Logout or Go Back buttons on top.

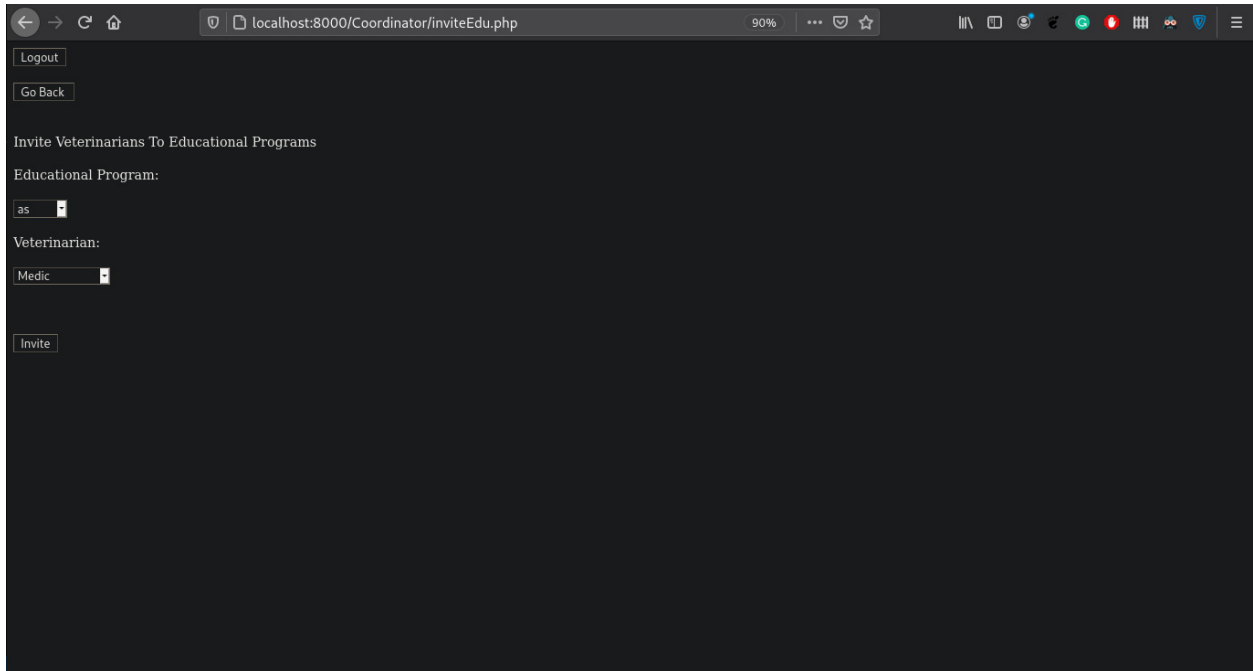
## **Send Emails to Visitors**

A screenshot of a web browser window displaying a form titled "Send E-mail to Visitors". The browser's address bar shows "localhost:8000/Coordinator/sendEmail.php". The form includes a "Logout" button at the top left, followed by a "Go Back" button. Below these, the title "Send E-mail to Visitors" is displayed. The form contains three input fields: "Event:" with a dropdown menu showing "group1", "Subject:" with a text input field, and "Message:" with a large text area. A "Send" button is located at the bottom left of the form.

Picture 6.3.9

If you want to send multiple emails to multiple users in a short time, click on the Send Emails To Visitors button on the main screen. After clicking, you will see the screen on Picture 6.3.9. Select a group tour from the dropdown menu, write a subject and a message. All visitors with a valid email will receive an email from [datanimal.bilkent@gmail.com](mailto:datanimal.bilkent@gmail.com) with the specified subject and message. This way, you can inform visitors about the events they are attending easily.

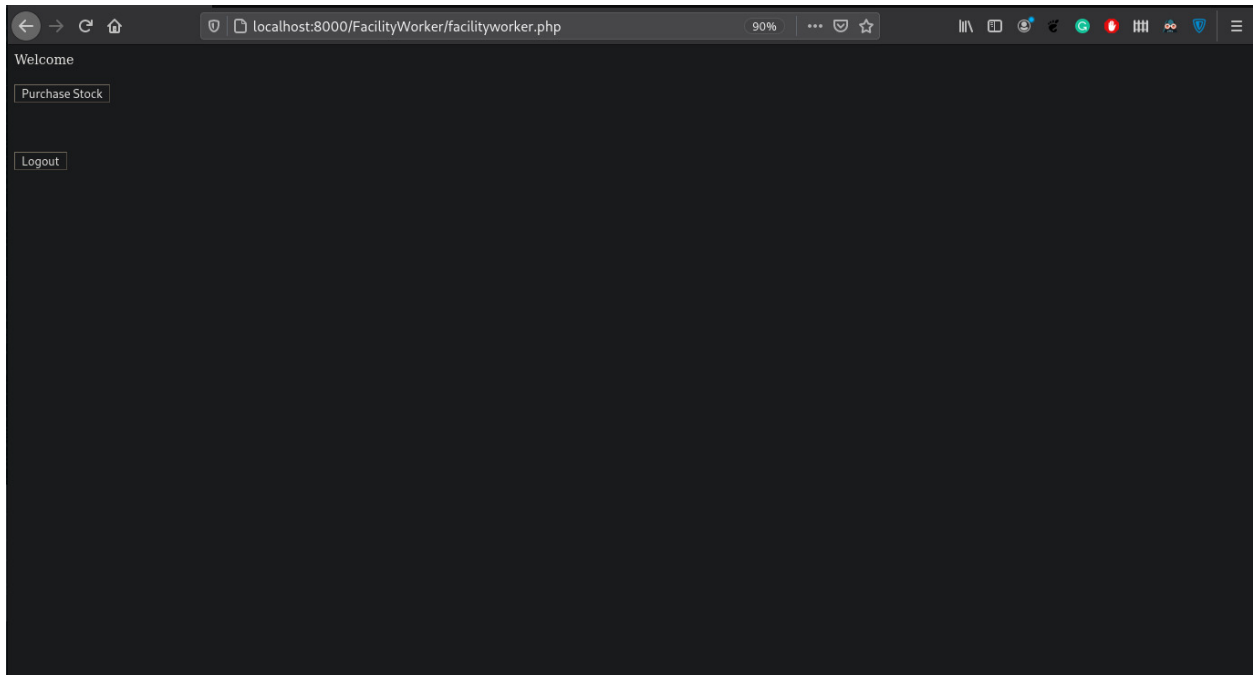
## **Invite Veterinarians to Educational Programs**



Picture 6.3.10

In order to invite a veterinarian to an educational program, click on the Invite Veterinarians To Educational Programs button and you will see a screen similar to Picture 6.3.10. From this page, select a program and a veterinarian using related dropdown menus. Later click on the Invite button. The veterinarian you have selected will receive an invitation for the educational program you selected.

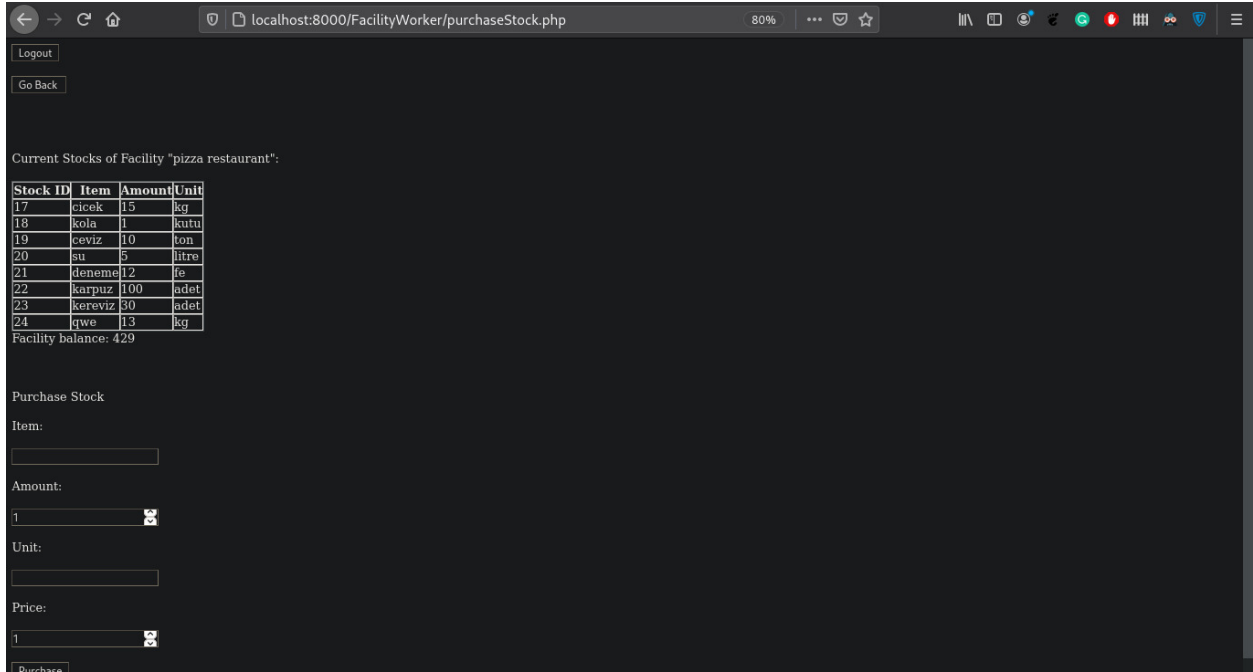
## 6.4. Facility Worker



Picture 6.4.1

If you log in successfully as a Facility Worker, you will see the screen shown in Picture 6.4.1. This is the main screen for Facility Worker. You can choose to Purchase Stock or Logout using this screen.

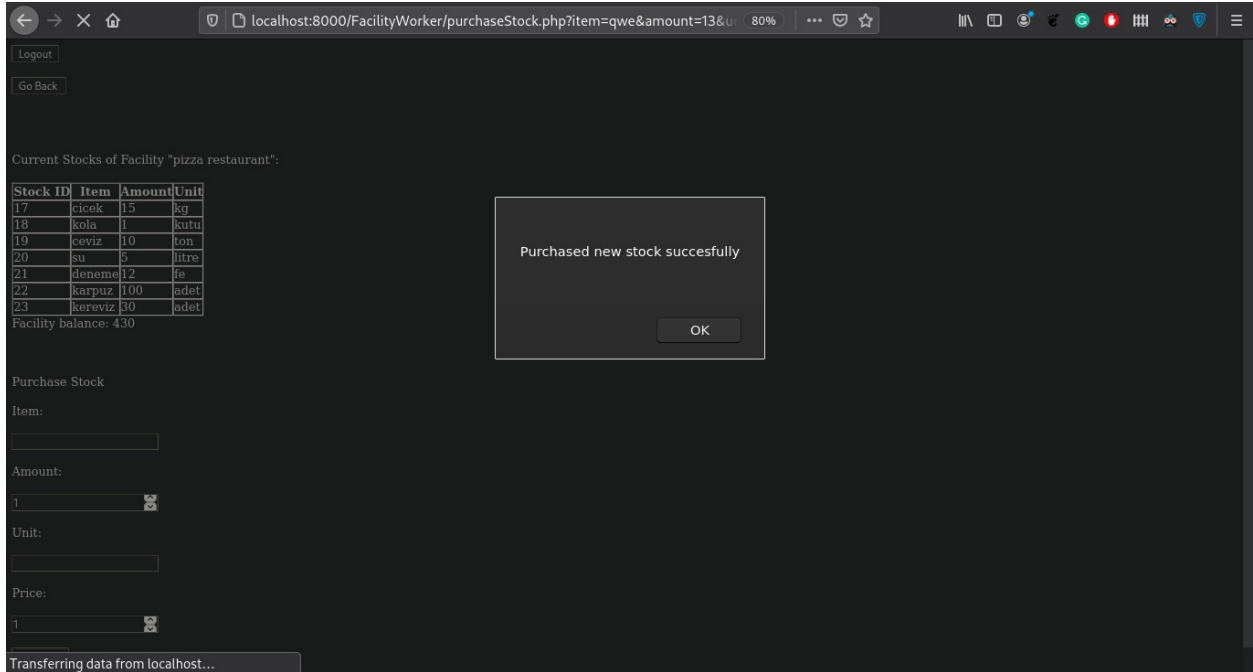




Picture 6.4.2

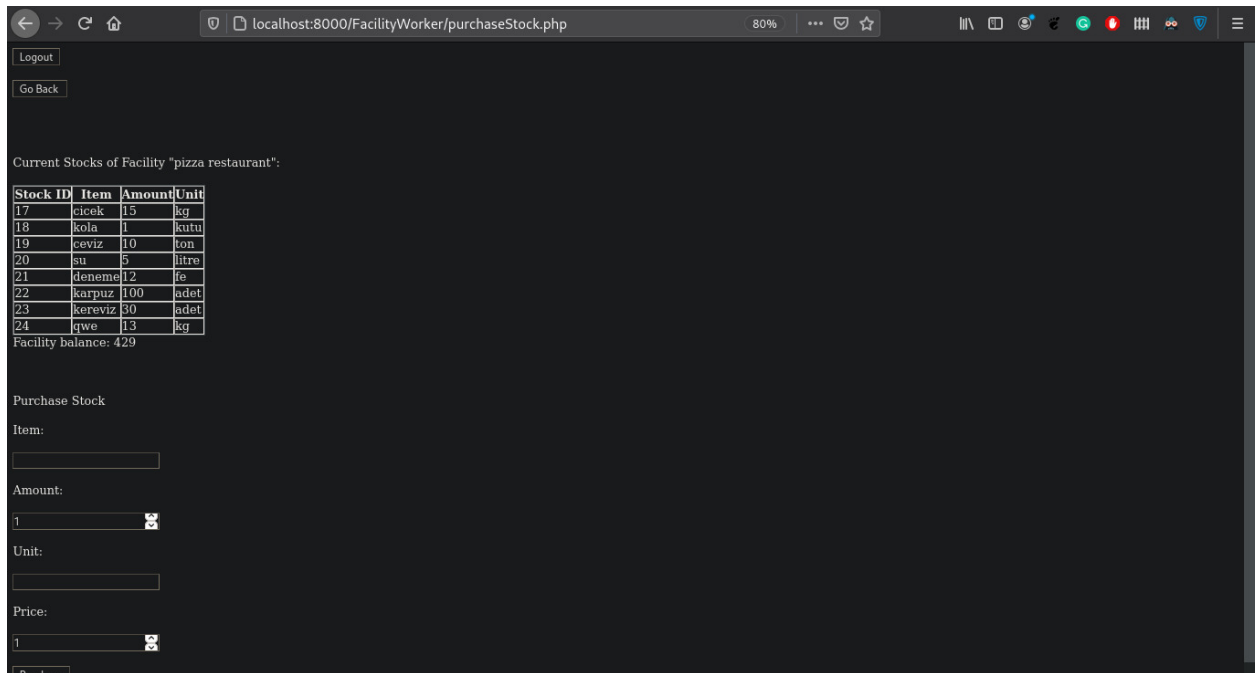
If you click on the Purchase Stock button on Facility Worker main screen, you will see the screen shown on Picture 6.4.2. In this page, you can see the current stocks of the facility you are working in and the current money of your facility.

You can purchase stock from this screen. After filling all the fields with necessary information and clicking the Purchase button , you will purchase the stock with the values you specified. However, if your facility does not have enough money, you will get a warning message and purchase will not occur. Also if you leave any fields empty, you will get a warning message.



Picture 6.4.3

If you fill all the fields and your facility has enough money, you will see a message indicating purchase was successful. After clicking the OK button, you can see the new stocks with newly purchased stock and updated Facility balance as shown in Picture 6.4.4.

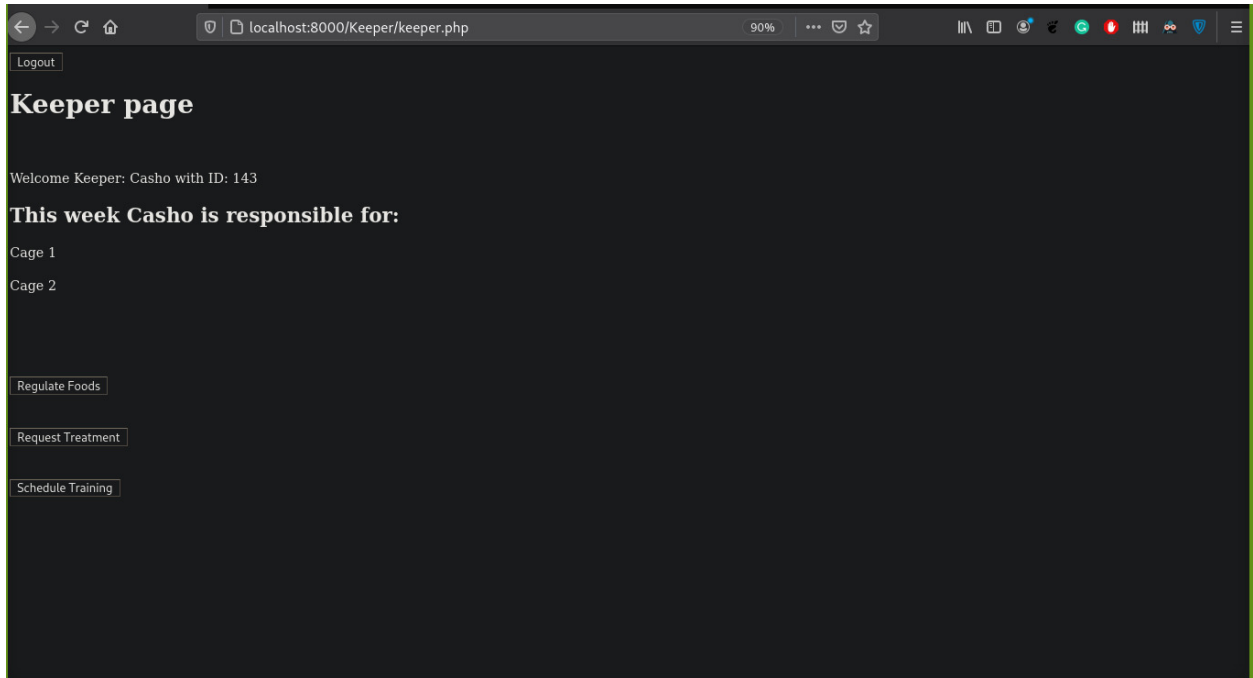


Picture 6.4.4

Also you can go back to the main page or Logout using the buttons on top.

## 6.5. Keeper

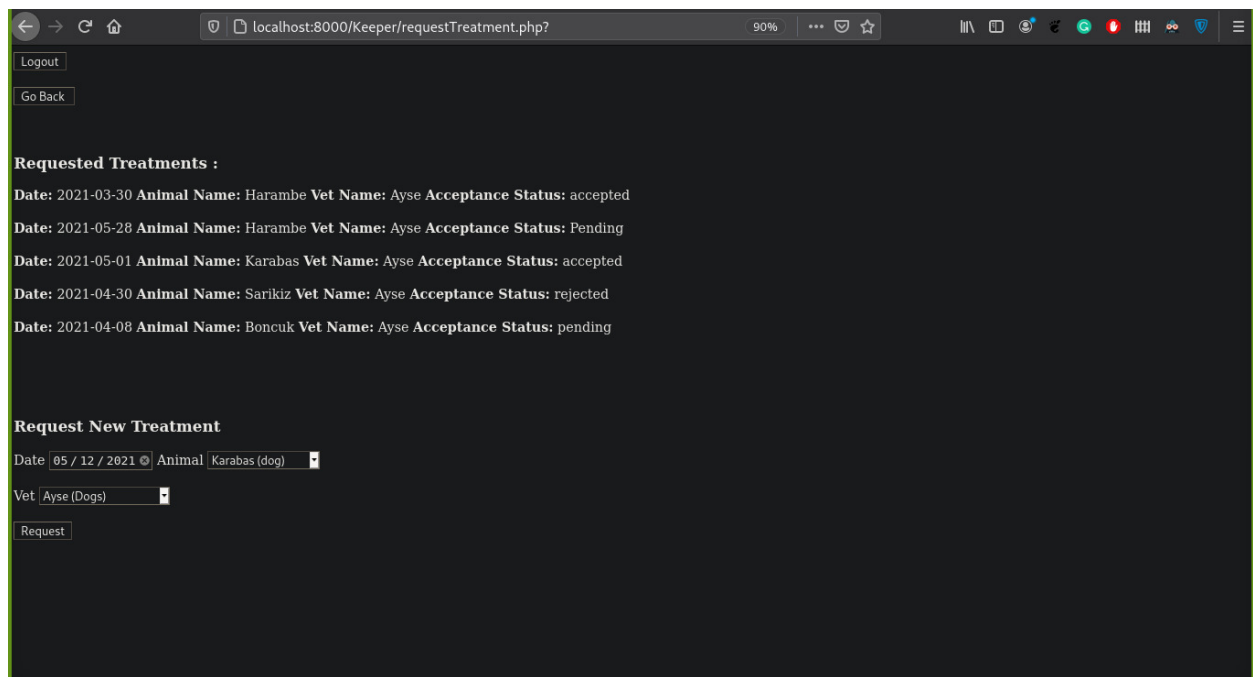
### Keeper Home Page



Picture 6.5.1

If you log in successfully as a Keeper, you will see the screen shown in Picture 6.5.1. This is the main screen for a Keeper. Under the “This week ... is responsible for:” header, the cages that have been assigned to the user can be seen. Under that there are buttons that can be clicked to regulate food, to request treatment or to schedule training. The “Logout” button can be clicked to go back to the login screen.

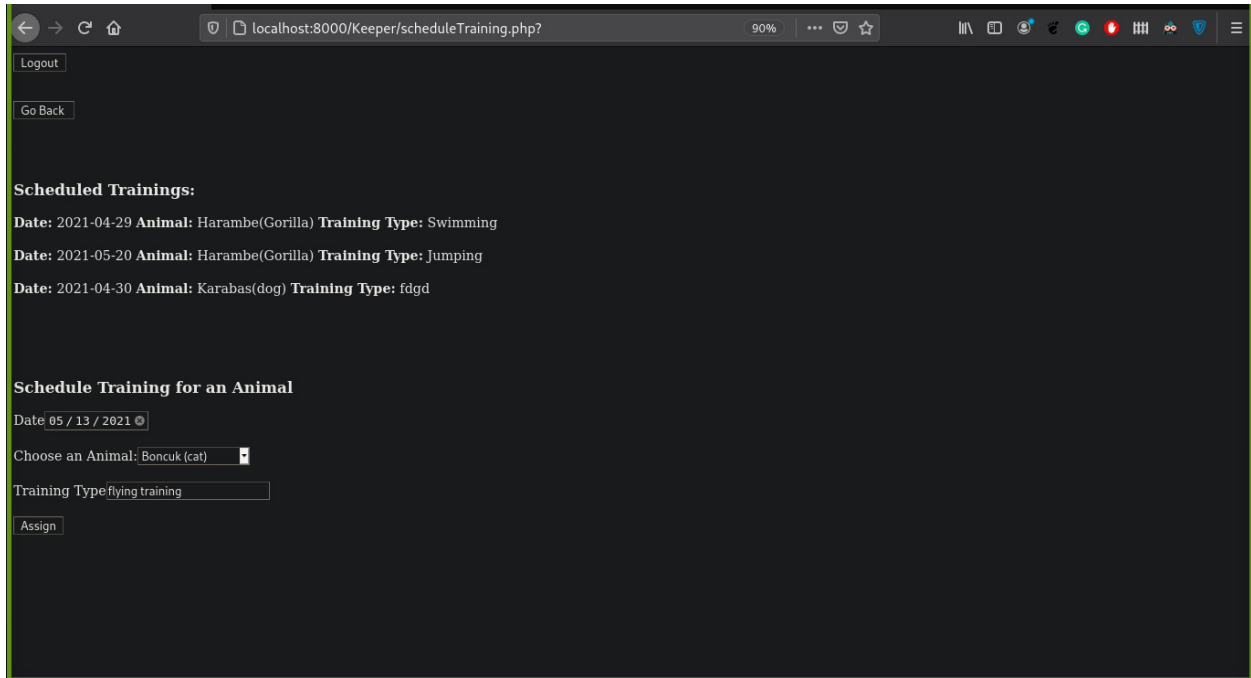
### **Request Treatment**



Picture 6.5.2

When the keeper clicks the “Request treatment” button on his/her home page, they are directed here. Under the “Requested Treatments:” header, the treatments that have already been requested by this user can be seen. The user can request a new treatment for an animal that lives in a cage assigned to this user by specifying the date, the requested veterinarian and the animal by using the dropdown options. When they click submit, the treatment request would be inserted into the database.

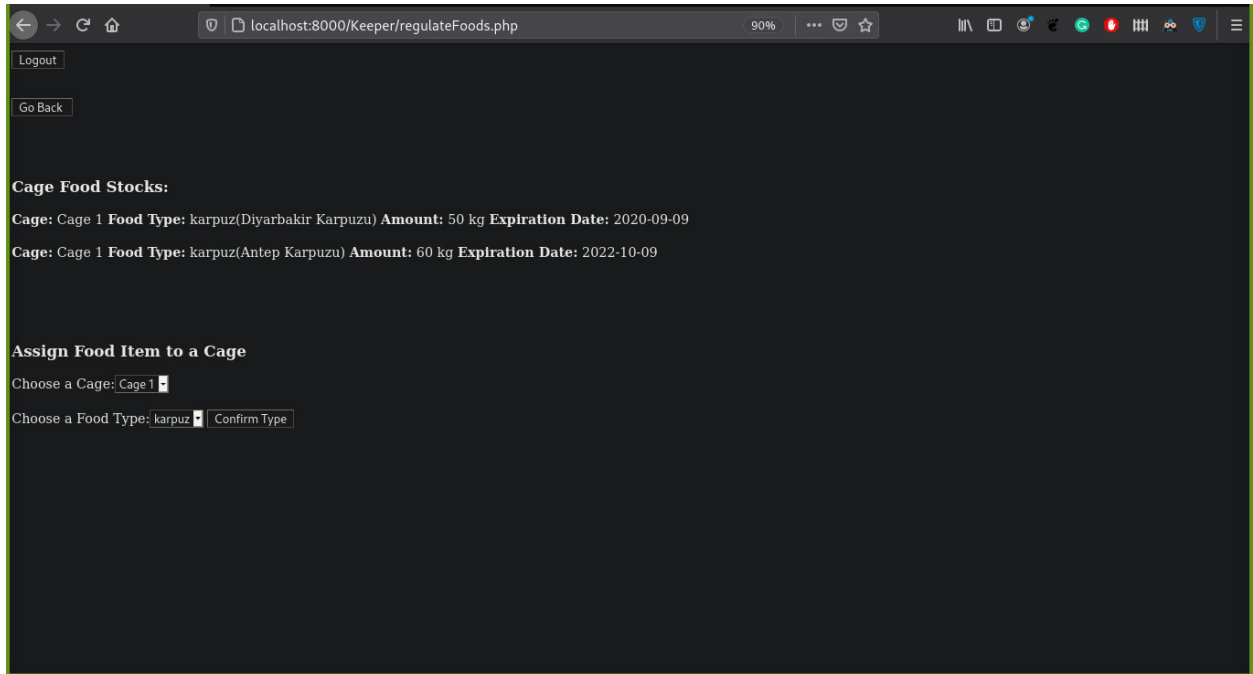
### **Schedule Training**



Picture 6.5.3

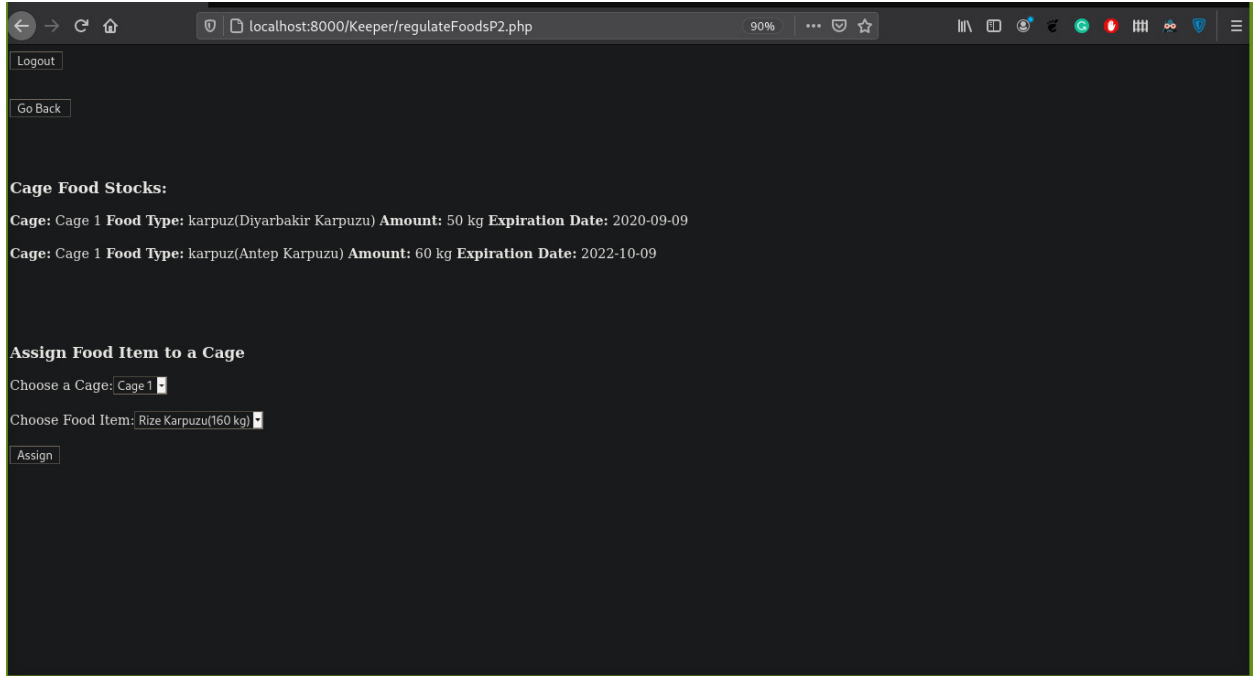
When the keeper clicks the “Schedule Training” button on his/her home page, they are directed here. Under the “Scheduled Trainings:” header, the trainings that have already been scheduled by this user can be seen. The user can request a new treatment for an animal that lives in a cage assigned to this user by specifying the date, the training type and the animal by using the dropdown options. When they click submit, the scheduled training would be inserted into the database.

### **Regulate Foods**



Picture 6.5.4

When the keeper clicks the “Regulate Foods” button on his/her home page, they are directed here. Under the “Cage Food Stocks:” header, the food stock that have already been assigned by this user can be seen. The user can assign a new food stock to a cage that has been assigned to this user by specifying the cage and then the food type, in the next page, food items that are of the type selected will be shown. When the user clicks “Confirm type” a new page that contains food items that are of the type selected will be shown.

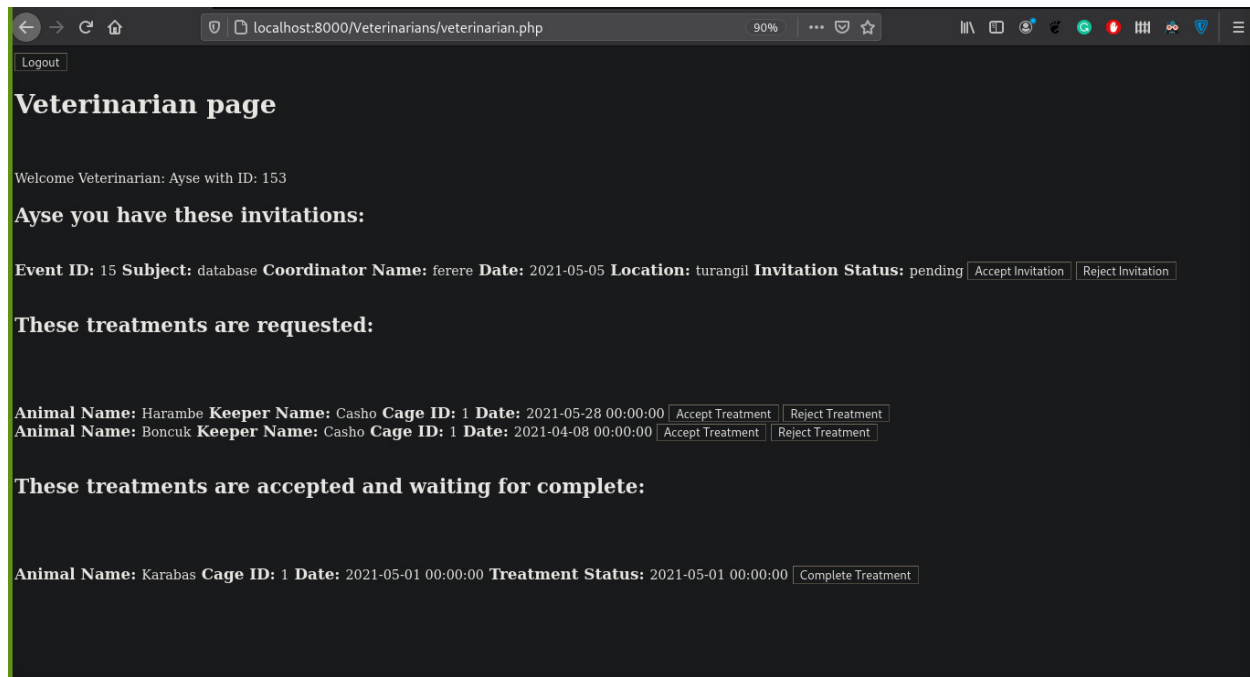


Picture 6.5.5

In this page, the user selects the food item that are part of the type selected(karpuz, muz etc.) from a dropdown menu that shows the unassigned foods from the database. On the dropdown menu, the food item name and its amount is shown (ex. Diyarbakir Karpuzu(45 kg)). When the user clicks the “Assign” button, the specified food item is assigned to the specified cage.



## 6.6. Veterinarian

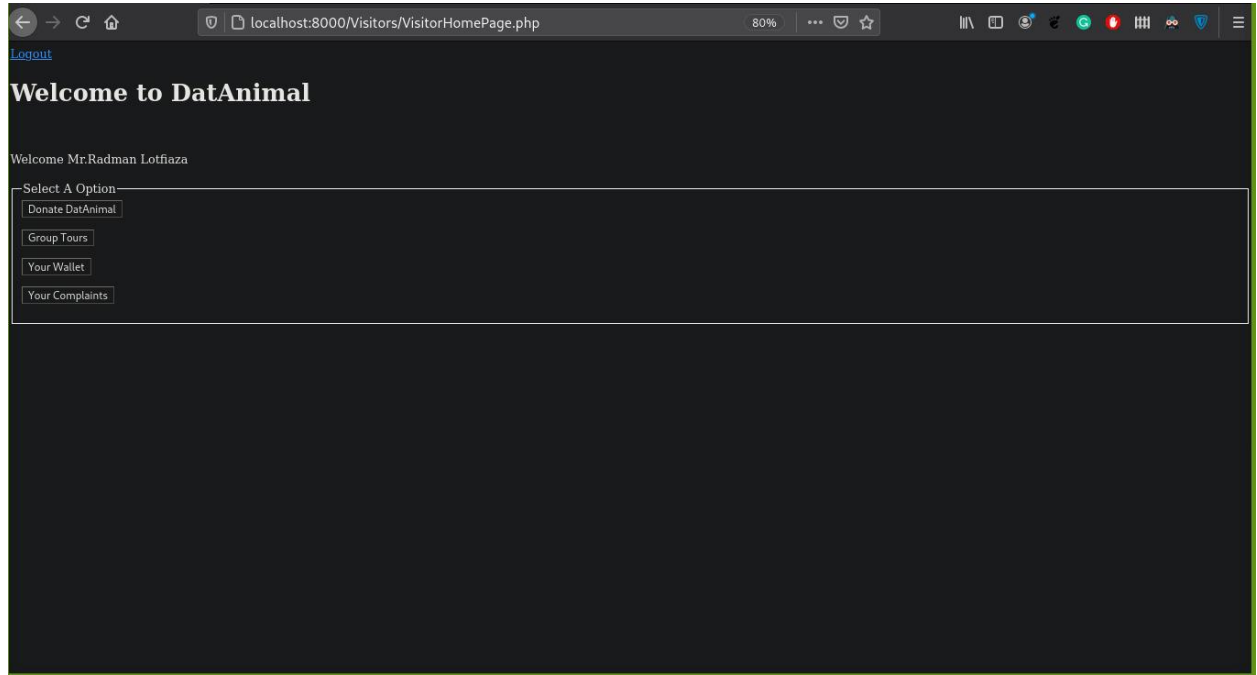


Picture 6.6.1

If you log in successfully as a Veterinarian, you will see the screen shown in Picture 6.6.1. This is the main screen for a Veterinarian. Under the “... you have these invitations:” header, the invitations that have been sent to this user can be seen. Besides every invitation line, there are accept/reject invitation buttons that allow the user to either accept or reject invitations. Under the “These treatments are requested:” header, the pending treatment requests made to this user are shown. Besides every request, there are accept/reject buttons. These buttons allow the user to either accept or reject a treatment request. Under the “These treatments are accepted and waiting for complete:” header, the already accepted but still incomplete treatments are shown. Besides every request, there is a complete treatment button. When pressed, the requests becomes complete and are no longer shown. The “Logout” button can be clicked to go back to the login screen.

## 6.7. Visitor

### 6.7.1. Visitor Home page



After the login page if your username and password are matched with visitors you will be faced with this page. In other words, this page is the Visitor's home page where visitors by button which are shown are able to donate to a conservation organization, attend a group tour, charge their wallet and complain about anything related to the zoo.

## 6.7.2. Donate

localhost:8000/Visitors/Donate.php?

[Home Page](#)

### We Will be Happy if You Donate Our Organization

Your Credit: 1472\$

**Your Past Donations**

Event Name	location	Date of Organization	Total of your Donation
org1	entrance	2021-05-05	100 \$
asd	qwe	2021-05-12	200 \$

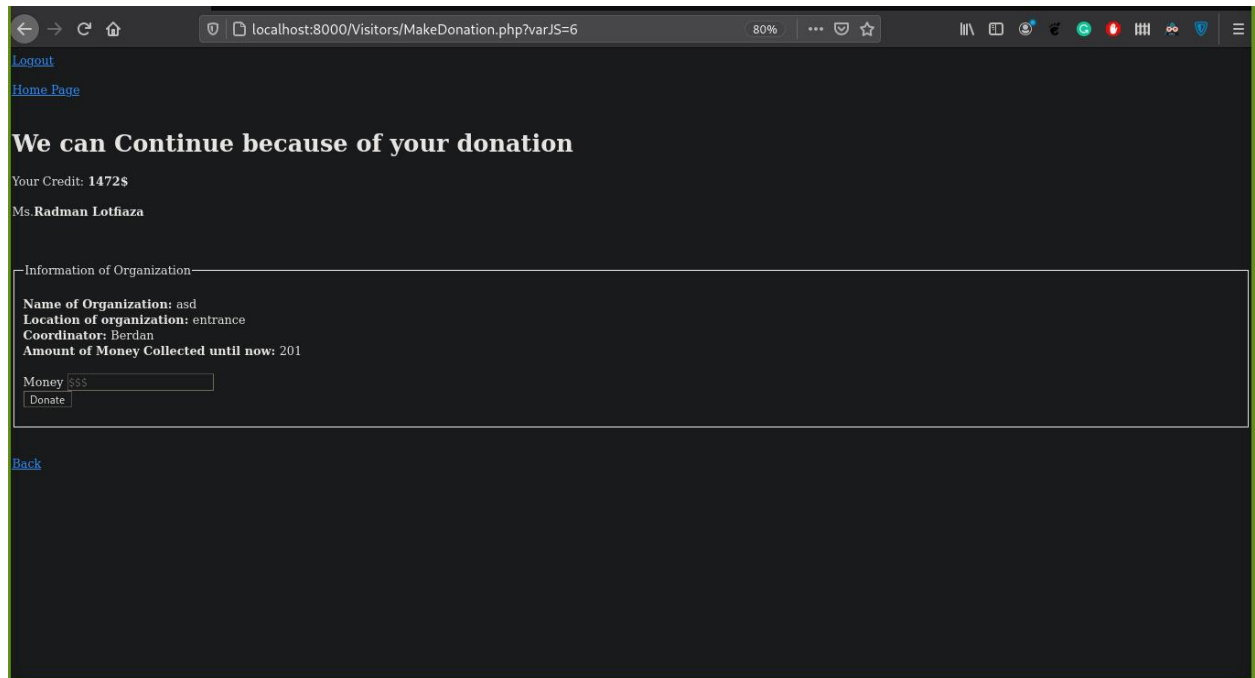
**Organizations For Your Donation**

Organization

Event Name	location	Date of Organization	Total Donated Money	Donate
org1	entrance	2021-05-05	100 \$	<input type="button" value="Donate"/>
asd	entrance	2021-05-12	201 \$	<input type="button" value="Donate"/>
asd	qwe	2021-05-12	200 \$	<input type="button" value="Donate"/>
radman	entrance	2021-05-12	200 \$	<input type="button" value="Donate"/>
org1	entrance	2021-05-05	0 \$	<input type="button" value="Donate"/>
gunes	gunesinev	2021-05-13	0 \$	<input type="button" value="Donate"/>

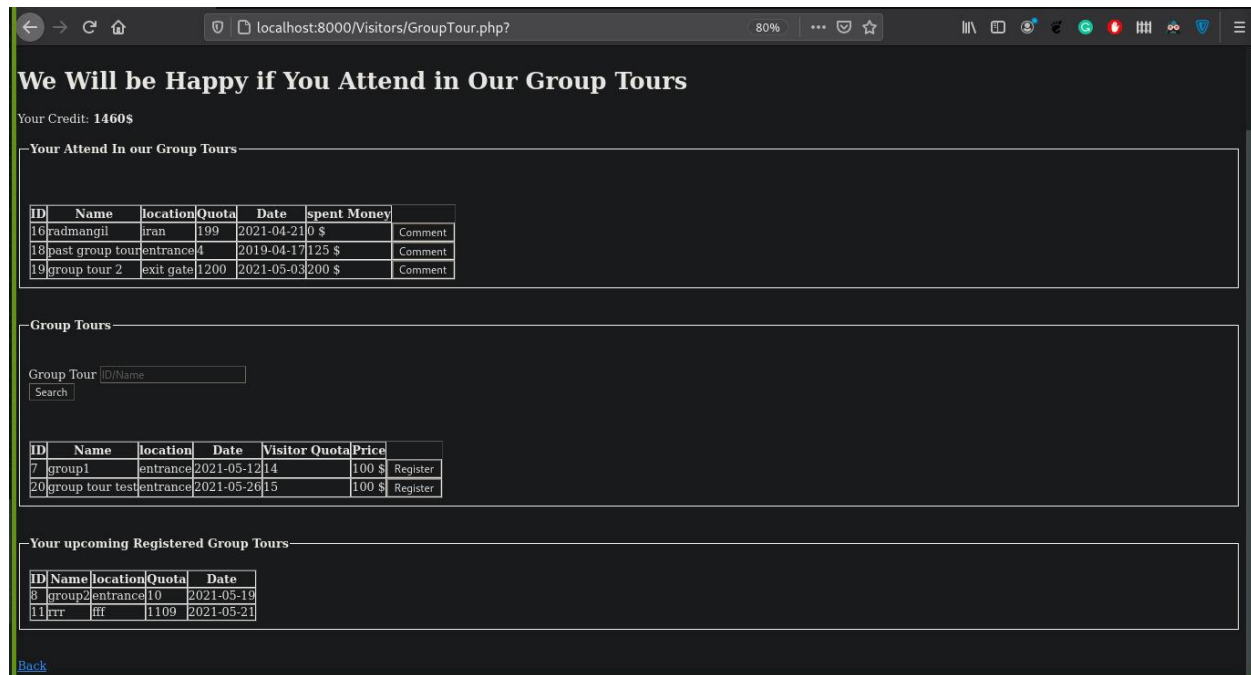
[Back](#)

After pressing DatAnimal you follow up with the above page which shows all conservation organizations and those you donated. In your past donations you are able to see the name, location and date of the conservation organization also, and the amount of money you already donate to that organization. The second table shows all conservation organization information and the amount of money which has been donated by people until now. There is a search button where you are able to search the name or ID of the organization and it will represent the organization with that name or ID. Furthermore, for each row there is a Donation Button which By pressing it you will be faced with the blow page.



In this page you are provided with the information about the conservation organization which you have already selected to donate. You can insert the amount of money which you want to donate and if you have that much money in your wallet you are able to pay, otherwise you will be faced with an error.

### 6.7.3. Attend Group Tour



The screenshot shows a web browser at the URL `localhost:8000/Visitors/GroupTour.php?`. The page has a dark theme and displays the following content:

## We Will be Happy if You Attend in Our Group Tours

Your Credit: **1460\$**

### Your Attend In our Group Tours

ID	Name	location	Quota	Date	spent Money	
16	radmangil	iran	199	2021-04-21	0 \$	Comment
18	past group tour	entrance	4	2019-04-17	125 \$	Comment
19	group tour 2	exit gate	1200	2021-05-03	200 \$	Comment

### Group Tours

Group Tour

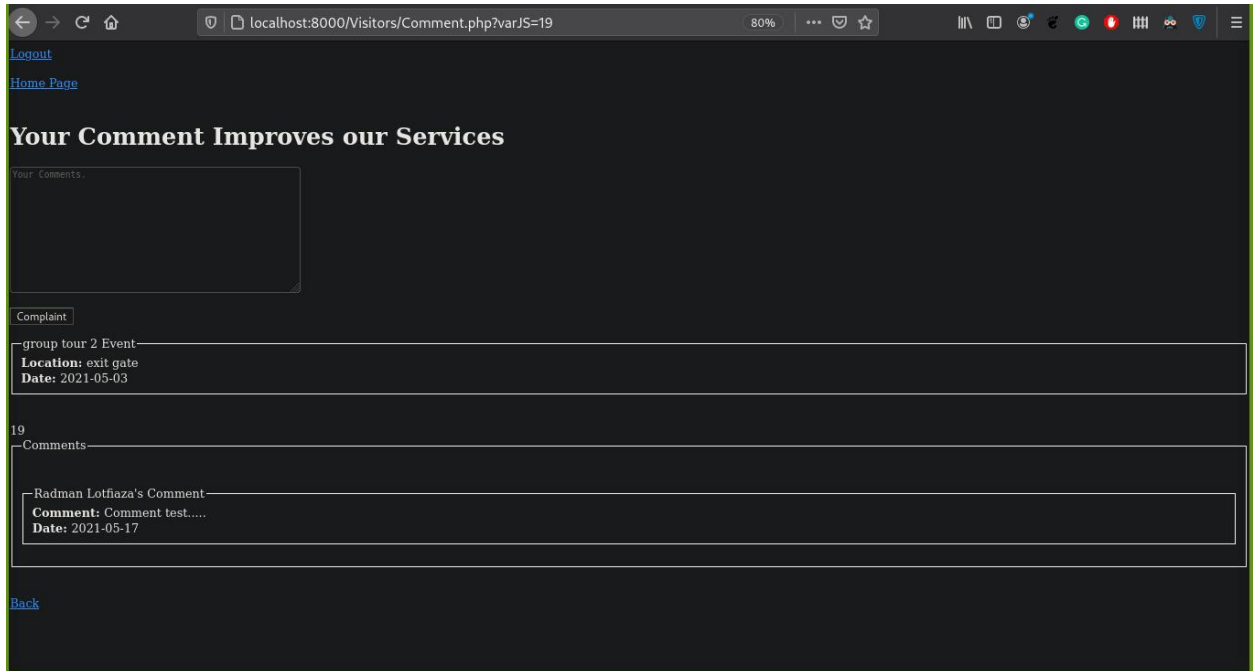
ID	Name	location	Date	Visitor Quota	Price	
7	group1	entrance	2021-05-12	14	100 \$	Register
20	group tour test	entrance	2021-05-26	15	100 \$	Register

### Your upcoming Registered Group Tours

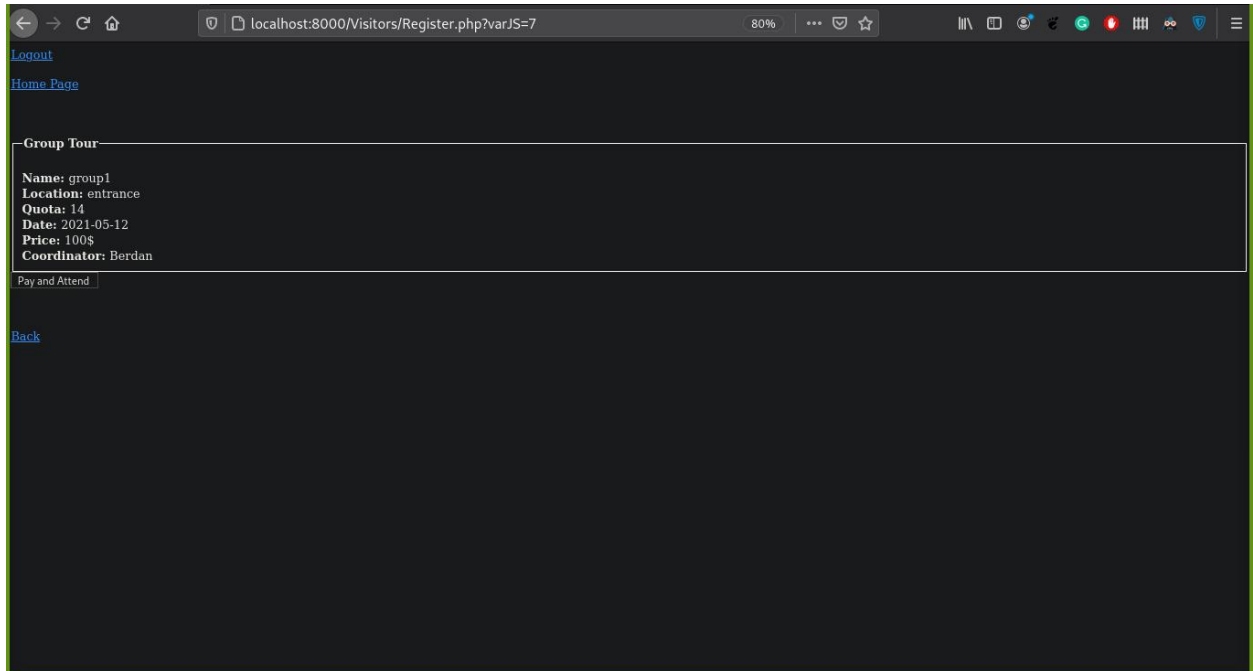
ID	Name	location	Quota	Date
8	group2	entrance	10	2021-05-19
11	rrr	fff	1109	2021-05-21

[Back](#)

After pressing the attend group tour button from visitor home page you will be faced with the above page. First table shows all group tours which you attended in the past. In that table there is information about those group tours and the amount of money you spent in there. By the Comment button next to each group tour you will be faced with the blow page.



In the above page you are able to leave a comment and see the other visitors' comment about that group tour which you selected from the first table of the attend group tours page. Second table of the attend group tours page you are provided with all upcoming group tours which you have are not registered yet. Therefore, by pressing the register button for each of them you are faced with the below page.



This page will show the information about group tours which you select and by pressing the Pay and Attend button if you have enough credit its price will be decreased from your credit and it will be shown in the third table in group tours table.

In the third table you are able to see the upcoming group tours which you have already registered for them.

## 6.7.4. Wallet

localhost:8000/Visitors/ChargeWallet.php? 80%

[Logout](#)  
[Home Page](#)

### Charge Your DatAnimal Wallet

You can charge your Wallet from here Ms.Radman Lotfiaza

Your Credit: **1360\$**

#### Your Card Information

Amount of Money

Your Card Number

Data Of Issue

CVV

From this page you are able to charge your account by your bank card. You will provide your credit card information and amount of money which you want to add to your DatAnimal wallet. After pressing charge you will be faced with a warning (Are you sure?) and by pressing the yes button specified amount of money will transfer from your bank account to your wallet.



### 6.7.5. Complaint

The screenshot shows a web browser window with the address bar displaying 'localhost:8000/Visitors/Complaint.php?'. The page has a dark background. At the top left, there are links for 'Logout' and 'Home Page'. Below these, there is a 'Topic' label and a text input field. A message below the input field says 'We will be happy to hear from you.' Below this is a 'Complaint' label and a large text area for writing a complaint. At the bottom, there is a section titled 'Complaints and Answers' which contains two entries. Each entry shows a complaint from a visitor and a response from a coordinator. The first entry is from 'Berdan2' with the topic 'Test topic', complaint 'Test message', and date '2021-05-05'. The response is from 'Coordinator Berdan' with the response 'replying' and date '2021-05-17'. The second entry is from 'Turan' with the topic 'Test topic2', complaint 'Test message2', and date '2021-05-06'. The response is from 'Coordinator Berdan' with the response 'test reply' and date '2021-05-17'.

Logout

Home Page

Topic

We will be happy to hear from you.

Complaint

Complaints and Answers

Berdan2's Complaint

Topic: Test topic

Complaint: Test message

Date: 2021-05-05

Coordinator Berdan's Respond

Respond: replying

Date: 2021-05-17

Turan's Complaint

Topic: Test topic2

Complaint: Test message2

Date: 2021-05-06

Coordinator Berdan's Respond

Respond: test reply

Date: 2021-05-17

Turan's Complaint

Through the above page you are able to see all complaints related to the zoo and their answer. You can complain from the text box which is provided in this page. For each complaint which you leave you have to choose a topic. Therefore, Coordinators can see the complaints and answers to them. In addition you can see the name of visitors who leave a complaint and coordinators who answer.