

Turtleback Zoo Online Application Project

Deliverable 3

Team-10:

Sathvika Karri (sk3568)

Sravya Kodavatiganti (sk3568)

Bhanu Sai Nikhil Vinnakota (bv258)

Goal of the current phase of the project:

The main aim of Phase 3 is to execute and document the Asset Management application program designed for Turtleback Zoo. This entails crafting SQL commands to establish tables, filling them with example data, and constructing a user interface driven by a menu. The primary goals include achieving flawless execution, producing a user guide, and overcoming any challenges that arise during implementation. The database system is intended to enable the zoo to effectively oversee its assets, daily operations, and reporting capabilities.

Databases Involved in the TurtlebackZoo Management System:

The structure of the database comprises several tables, each serving a specific purpose:

- **Animals:** Holds details about zoo animals, including ID, name, species, status, food cost, veterinarian ID, and animal care specialist ID.
- **Buildings:** Stores information about zoo buildings, such as ID, name, type, and capacity.
- **Attractions:** Contains details about zoo attractions, including ID, name, type, building ID, show time, and ticket price.
- **Employees:** Records information about zoo staff, including ID, name, role, and phone number.

- **Employee_Hourly_Wages:** Manages data on the hourly wages of zoo employees, including ID, date, hours worked, and hourly rate.
- **Concessions:** Stores information about zoo concessions, including ID, name, type, and revenue.
- **Attendance:** Records information on zoo attendance and revenue, including date, number of attendees, and total revenue.

To ensure data integrity and facilitate retrieval, the database schema incorporates primary keys, secondary keys, and foreign keys. The SQL commands for table creation and insertion can be found in the turtlebackzoo.sql file.

Challenges Faced During this Phase:

Encountering Database Connection Issues: Achieving a smooth connection between your PHP project and the MySQL database within XAMPP can be a nuanced process, requiring careful attention to configuration parameters and potential authentication challenges. Establishing a seamless connection may involve adjusting settings to align with the specifics of your development environment.

Navigating PHP Version Compatibility Challenges: When your production server operates on a PHP version different from your XAMPP setup, compatibility issues may arise, necessitating thoughtful resolution. It becomes crucial to address any disparities between PHP versions to ensure that your PHP project functions consistently across both your local development environment and the production server.

Users Guide:

The home page serves as the central hub for accessing various functionalities within the Zoo Management Application. The application is designed to streamline and enhance the management of zoo assets, monitor daily activities, and generate insightful reports.

- **Navigation Bar**

Located at the top of the home page, the navigation bar offers quick access to different modules of the application. The available options are:

- **Asset Management:** Click on this option to handle information related to animals, buildings, attractions, employees, concessions, employee hourly wages, and attendance.
- **Daily Zoo Activity:** Navigate to this section for real-time monitoring and management of daily zoo operations, including attendance tracking, staff scheduling, and activity coordination.
- **Reports:** Access the reporting module to generate comprehensive reports based on various parameters, helping you analyze the performance and trends within the zoo.

- **Asset Management Page Options Overview**

- **Insert Animal:**

- Purpose: Add new animal information to the database.
- Procedure: Click to enter details such as ID, name, species, status, food cost, veterinarian ID, and animal care specialist ID.

- **Update/View Animal:**

- Purpose: Modify or review existing animal information.
- Procedure: Access to update details like status or view comprehensive information about animals.

- **Insert Building:**

- Purpose: Introduce new zoo buildings to the database.
- Procedure: Enter details such as ID, name, type, and capacity for the new building.

- **Insert Attraction:**

- Purpose: Add new attractions within the zoo.
- Procedure: Provide details like ID, name, type, associated building, show time, and ticket price for the new attraction.

- **Update/View Attractions:**

- Purpose: Modify or review existing attraction details.
- Procedure: Access to update information such as show time or view detailed data about attractions.

- **Add Employee:**
 - Purpose: Register new employees.
 - Procedure: Input details like ID, name, role, and phone number for the new employee.
- **Manage Employee:**
 - Purpose: Modify or review existing employee details.
 - Procedure: Access to update information such as role or view comprehensive data about employees.
- **Add Animal Species:**
 - Purpose: Introduce new animal species to the database.
 - Procedure: Enter details like species ID, name, and any additional relevant information.
- **Update/View Species:**
 - Purpose: Modify or review existing species details.
 - Procedure: Access to update information or view comprehensive data about animal species.

Important Notes:

- Updating Data: For options labeled "Update/View," you have the ability to both modify existing data and view detailed information.
- Data Integrity: Ensure accurate information entry for effective management and reporting.

These options collectively empower you to maintain an accurate and comprehensive record of zoo assets while facilitating seamless interaction with the underlying database.

- **Reports Page Options Overview:**
- **Revenue Report:**
 - Purpose: Generates a comprehensive report on the zoo's revenue.
 - Insights: Provides insights into the financial performance of the zoo over a specified period.
- **Species Report:**

- Purpose: Generates a report detailing information about various animal species in the zoo.
- Insights: Offers insights into the diversity and management of animal species within the zoo.
- **Five Best Days:**
 - Purpose: Identifies and presents information on the top five days with the highest attendance.
 - Insights: Helps understand peak attendance days for planning events or promotions.
- **Top Three Attractions:**
 - Purpose: Provides a report highlighting the three most popular attractions based on attendance.
 - Insights: Helps in optimizing resource allocation and improving visitor experience.
- **Report on Revenue/Total Attendance:**
 - Purpose: Generates a report comparing revenue and total attendance.
 - Insights: Offers insights into the correlation between revenue and attendance, aiding in strategic planning.

These reporting options are designed to provide you with a holistic view of the zoo's performance, allowing you to make informed decisions and enhance the overall management and visitor experience.

- **Daily Zoo activity page**

Purpose: Record the number of attendees and the revenue generated for a specific day's events.

- **Procedure:**
 - **Select Date:** Choose the date for which you want to record attendance and revenue. This ensures accurate tracking of daily activities.
 - **Enter Attendance:** Input the number of visitors who attended the zoo on the selected date. This includes both regular visitors and those attending special events.
 - **Enter Revenue:** Record the total revenue generated on the chosen date, considering ticket sales, concessions, and other sources.

SQL queries to create the tables in the database:

```
CREATE TABLE Buildings (
    BuildingID INT AUTO_INCREMENT PRIMARY KEY,
    Name VARCHAR(255),
    Purpose VARCHAR(255),
    Floors INT,
    SquareFootage INT
) ENGINE=InnoDB;
```

Tablename: Buildings

Primary keys: BuildingID

```
CREATE TABLE Employees (
    EmployeeID INT AUTO_INCREMENT PRIMARY KEY,
    Name VARCHAR(255),
    Address VARCHAR(255),
    StartDate DATE,
    EmployeeType VARCHAR(50),
    HourlyRate DECIMAL(10, 2)
) ENGINE=InnoDB;
```

Tablename: Employees

Primary keys: EmployeeID

```
CREATE TABLE Animals (
    AnimalID INT AUTO_INCREMENT PRIMARY KEY,
    Species VARCHAR(255),
    LocationBuilding VARCHAR(255),
    LocationCage VARCHAR(255),
    BirthYear INT,
    CurrentStatus VARCHAR(50)
) ENGINE=InnoDB;
```

Tablename: Animals

Primary keys: AnimalID

```
CREATE TABLE EmployeeTypes (
    EmployeeType VARCHAR(50) PRIMARY KEY,
    HourlyRate DECIMAL(10, 2),
    DegreeYear INT,
    SpeciesSpecialties VARCHAR(255)
) ENGINE=InnoDB;
```

Tablename: EmployeeTypes

Primary keys: EmployeeType

```
CREATE TABLE Species (
    SpeciesID INT AUTO_INCREMENT PRIMARY KEY,
    Population INT,
    MonthlyFoodCost DECIMAL(10, 2),
    VeterinarianID INT,
    AnimalCareSpecialistID INT,
    TrainerID INT
) ENGINE=InnoDB;
```

Tablename: Species

Primary keys: SpeciesID

```
CREATE TABLE Attractions (
    AttractionID INT AUTO_INCREMENT PRIMARY KEY,
    Name VARCHAR(255),
    BuildingID INT,
    ShowsPerDay INT,
    TicketPrice DECIMAL(10, 2)
) ENGINE=InnoDB;
```

Tablename: Attractions

Primary keys: AttractionID

```
CREATE TABLE TicketPrices (
    TicketType VARCHAR(50) PRIMARY KEY,
    Price DECIMAL(10, 2),
    EffectiveDate DATE
) ENGINE=InnoDB;
```

Tablename: TicketPrices

Primary keys: TicketType

```
✓ CREATE TABLE AttractionSpecies (
    AttractionID INT,
    SpeciesID INT,
    PRIMARY KEY (AttractionID, SpeciesID),
    FOREIGN KEY (AttractionID) REFERENCES Attractions(AttractionID),
    FOREIGN KEY (SpeciesID) REFERENCES Species(SpeciesID)
) ENGINE=InnoDB;
```

Tablename: AttractionSpecies

Primary keys: AttractionID, SpeciesID

Foreign keys: AttractionID, SpeciesID referencing to the primary keys from the tables Attractions and Species respectively.

```
✓ CREATE TABLE Concessions (
    ConcessionID INT AUTO_INCREMENT PRIMARY KEY,
    Name VARCHAR(255),
    DailyRevenue DECIMAL(10, 2)
) ENGINE=InnoDB;
```

Tablename: Concessions

Primary keys: ConcessionID

```
CREATE TABLE Attendance (
    AttendanceID INT AUTO_INCREMENT PRIMARY KEY,
    AttendeeType VARCHAR(50),
    TicketsSold INT,
    RevenueEarned DECIMAL(10, 2),
    Date DATE
) ENGINE=InnoDB;
```

Tablename: Attendance

Primary keys: AttendanceID

```
CREATE TABLE Shows (
    ShowID INT AUTO_INCREMENT PRIMARY KEY,
    AttractionID INT,
    Date DATE,
    TicketsSold INT,
    RevenueEarned DECIMAL(10, 2)
) ENGINE=InnoDB;
```

Tablename: Shows

Primary keys: ShowsID

```
CREATE TABLE EmployeeAttraction (
    EmployeeID INT,
    AttractionID INT,
    PRIMARY KEY (EmployeeID, AttractionID),
    FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID),
    FOREIGN KEY (AttractionID) REFERENCES Attractions(AttractionID)
) ENGINE=InnoDB;
```

Tablename: EmployeeAttraction

Primary keys: EmployeeID, AttractionID

Foreign keys: EmployeeID, AttractionID referencing to the primary keys from the tables Employees and Attractions respectively.

SQL queries to populate the tables in the database:

1. Inserting values into Buildings table:

```
INSERT INTO Buildings (Name, Purpose, Floors, SquareFootage)
VALUES ('Building1', 'Animal Shelter', 2, 2000),
       ('Building2', 'Attraction', 3, 3000),
       ('Building3', 'Maintenance', 1, 1000);
```

Three entries are added to the building table, each of which represents a different building by its name, purpose, number of floors and square meters.

2. Inserting values into Employees table:

```
INSERT INTO Employees (Name, Address, StartDate, EmployeeType, HourlyRate)
VALUES ('John Doe', '123 Main St', '2023-01-01', 'Maintenance', 15.00),
       ('Jane Smith', '456 Oak St', '2023-01-15', 'Animal Care Specialist', 18.00),
       ('Bob Johnson', '789 Pine St', '2023-02-01', 'Customer Service', 12.50);
```

Three records are added to the Employees table, each representing an employee with a name, address, start date, employee type, and hourly wage.

3. Inserting values into EmployeeTypes table:

```
INSERT INTO EmployeeTypes (EmployeeType, HourlyRate, DegreeYear, SpeciesSpecialties)
VALUES ('Maintenance', 15.00, NULL, NULL),
       ('Animal Care Specialist', 18.00, 4, 'Big Cats, Elephants'),
       ('Customer Service', 12.50, NULL, NULL);
```

Three entries are added to the EmployeeTypes table, each representing an employee type along with hourly wage, year of degree required, and sport.

4. Inserting values into Animals table:

```
INSERT INTO Animals (Species, LocationBuilding, LocationCage, BirthYear, CurrentStatus)
VALUES ('Lion', 'Building2', 'Cage1', 2018, 'Healthy'),
       ('Elephant', 'Building2', 'Cage2', 2010, 'Under Medical Care'),
       ('Giraffe', 'Building2', 'Cage3', 2015, 'Maternal Leave');
```

Three entries are added to the Animals table, each representing an animal of its own species, location in building, location in cage, year of birth, and current status.

5. Inserting values into species table:

```
INSERT INTO Species (Population, MonthlyFoodCost, VeterinarianID, AnimalCareSpecialistID, TrainerID)
VALUES (5, 2000.00, 1, 2, 3),
       (2, 3000.00, 1, 2, 3),
       (3, 2500.00, 1, 2, 3);
```

Three entries are added to the species table, each representing a species along with its population, monthly feed cost, vet ID, animal care specialist ID, and trainer ID.

6. Inserting values into Attractions table:

```
INSERT INTO Attractions (Name, BuildingID, ShowsPerDay, TicketPrice)
VALUES ('Monkey Show', 2, 3, 10.00),
       ('Big Cat Show', 2, 2, 12.50),
       ('Aquatic Show', 2, 4, 15.00);
```

Three entries are added to the attraction table, each representing an attraction with its name, building identifier, number of shows per day, and ticket price.

7. Inserting values into Show table:

```
INSERT INTO Shows (AttractionID, Date, TicketsSold, RevenueEarned)
VALUES (1, '2023-11-01', 50, 500.00),
       (2, '2023-11-01', 30, 375.00),
       (3, '2023-11-01', 40, 600.00);
```

Three records are added to the Show table, each representing a show with an attraction ID, date, tickets sold, and revenue earned.

8. Inserting values into EmployeeAttraction table:

```
INSERT INTO EmployeeAttraction (EmployeeID, AttractionID)
VALUES (2, 1),
       (3, 2),
       (1, 3);
```

Three records are added to the EmployeeAttraction table, each representing a relationship between an employee and an attraction.

9. Inserting values into TicketPrices table:

```
INSERT INTO TicketPrices (TicketType, Price, EffectiveDate)
VALUES ('Adult', 20.00, '2023-01-01'),
       ('Child', 10.00, '2023-01-01'),
       ('Senior', 15.00, '2023-01-01');
```

Three entries are added to the Ticket Prices Table, each representing a ticket type with its price and effective date.

10. Inserting values into AttractionSpecies table:

```
INSERT INTO AttractionSpecies (AttractionID, SpeciesID)
VALUES (1, 1),
       (2, 2),
       (3, 3);
```

Three entries are added to the AttractionTypes table, each representing a relationship between an attraction and a species.

11. Inserting values into Concession table:

```
INSERT INTO Concessions (Name, DailyRevenue)
VALUES ('Snack Stand', 150.00),
       ('Drink Stand', 100.00),
       ('Souvenir Shop', 200.00);
```

Three entries are added to the concession table, each representing a concession section and its name and daily revenue.

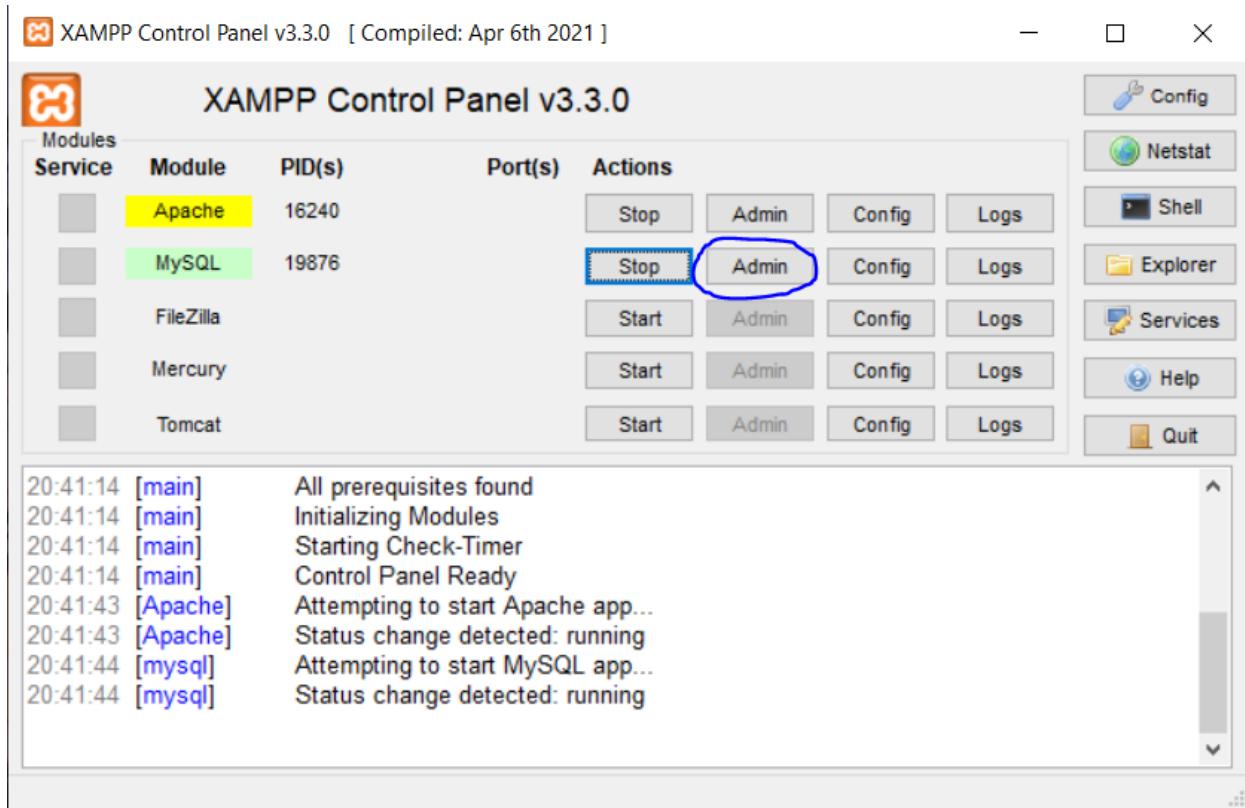
12. Inserting values into Attendance table:

```
INSERT INTO Attendance (AttendeeType, TicketsSold, RevenueEarned, Date)
VALUES ('Adult', 20, 400.00, '2023-11-01'),
       ('Child', 15, 150.00, '2023-11-01'),
       ('Senior', 10, 150.00, '2023-11-01');
```

Three entries are added to the attendees table, each representing a different type of attendee information, including number of tickets sold, revenue earned, and date.

PROGRAM USE:

1. Open Xampp Control Panel and click on “Start” for Apache and MySQL and click on “Admin” next to “Admin”.



2. pHpMyAdmin page will open up. Download the turtlebackzoo zip file and add that file in your system:

C: -> xmapp -> htdocs -> paste it there

The screenshot shows the pHpMyAdmin interface with the URL 'localhost / 127.0.0.1 / turtleback_zoo'. The left sidebar shows the database structure with tables like Buildings, Employees, EmployeeTypes, Animals, and Species. The main panel shows the results of an import operation:

```

Import has been successfully finished. 24 queries executed. (turtleback_zoo.sql)
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0004 seconds.)
CREATE TABLE Buildings ( BuildingID INT AUTO_INCREMENT PRIMARY KEY, Name VARCHAR(255), Purpose VARCHAR(255), Floors INT, SquareFootage INT ) ENGINE=InnoDB;
[Edit inline] [Edit] [Create PHP code]
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0010 seconds.)
CREATE TABLE Employees ( EmployeeID INT AUTO_INCREMENT PRIMARY KEY, Name VARCHAR(255), Address VARCHAR(255), StartDate DATE, EmployeeType VARCHAR(50), HourlyRate DECIMAL(10, 2) ) ENGINE=InnoDB;
[Edit inline] [Edit] [Create PHP code]
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0005 seconds.)
CREATE TABLE EmployeeTypes ( EmployeeType VARCHAR(50) PRIMARY KEY, HourlyRate DECIMAL(10, 2), DegreeYear INT, SpeciesSpecialties VARCHAR(255) ) ENGINE=InnoDB;
[Edit inline] [Edit] [Create PHP code]
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0012 seconds.)
CREATE TABLE Animals ( AnimalID INT AUTO_INCREMENT PRIMARY KEY, Species VARCHAR(255), LocationBuilding VARCHAR(255), LocationCage VARCHAR(255), BirthYear INT, CurrentStatus VARCHAR(50) ) ENGINE=InnoDB;
[Edit inline] [Edit] [Create PHP code]
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0013 seconds.)
CREATE TABLE Species ( SpeciesID INT AUTO_INCREMENT PRIMARY KEY, Population INT, MonthlyFoodCost DECIMAL(10, 2), VeterinarianID INT, AnimalCareSpecialistID INT, TrainerID INT ) ENGINE=InnoDB;

```

3. In pHpMyAdmin:

Go to home, go to Databases. In the zip file add the turtleback_sql in phpmyadmin.

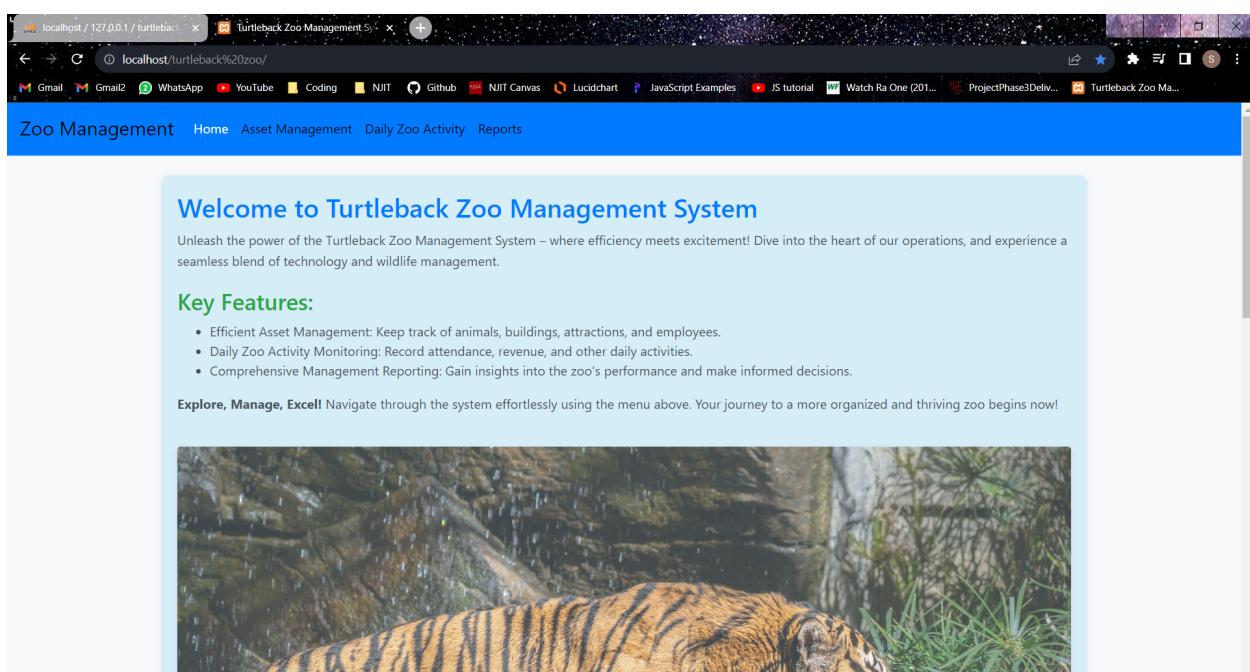
4. You will get all the tables on the left side.

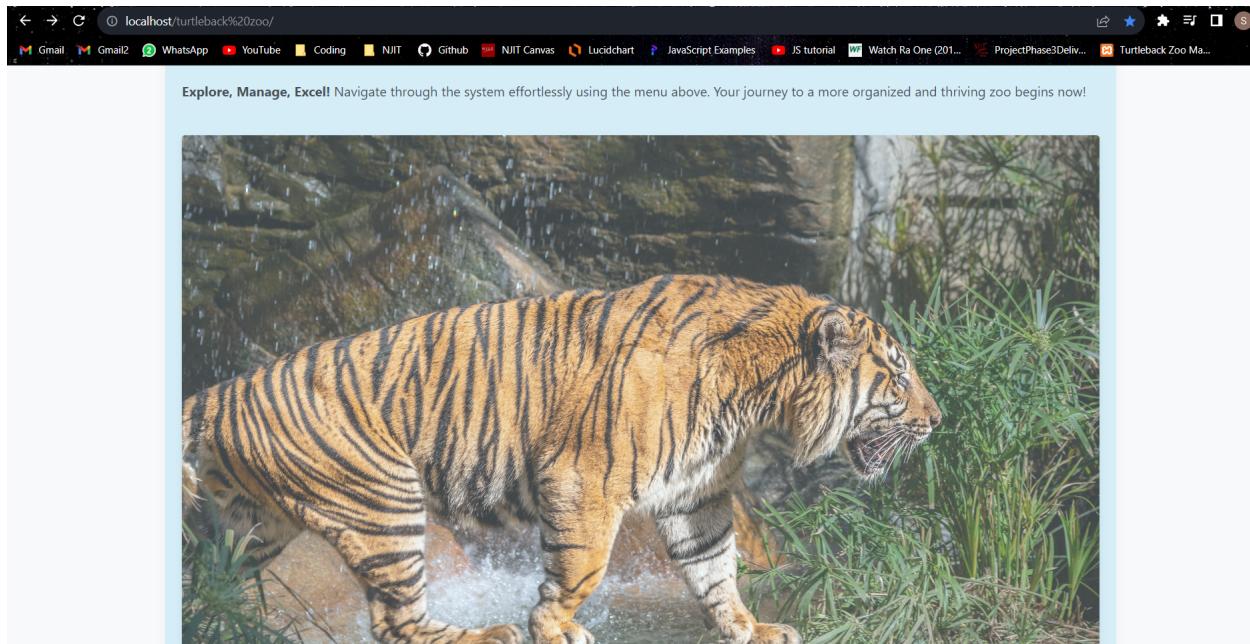
The screenshot shows the pHpMyAdmin interface with three tables displayed on the right and their corresponding structures on the left.

- Attendance:** Shows data for attendance types: Adult, Child, and Senior. The table has columns: AttendanceID, AttendeeType, TicketsSold, RevenueEarned, and Date.
- Buildings:** Shows data for buildings: Building1 (Animal Shelter), Building2 (Attraction), and Building3 (Maintenance). The table has columns: BuildingID, Name, Purpose, Floors, and SquareFootage.
- Attractions:** Shows data for attractions: Monkey Show, Big Cat Show, and Aquatic Show. The table has columns: AttractionID, Name, BuildingID, ShowsPerDay, and TicketPrice.

5. Open a new tab and type this URL: localhost/turtleback_zoo/

Or <http://localhost/turtleback%20zoo/>





About Turtleback Zoo

Welcome to Turtleback Zoo – a realm of wonder and conservation! Our commitment to wildlife preservation is as deep as our dedication to providing an unforgettable experience for visitors of all ages.

Embark on a Journey: From roaring lions to graceful dolphins, each exhibit tells a story of nature's marvels. At Turtleback Zoo, we don't just showcase animals; we celebrate life and inspire a passion for the wild.

Join the Adventure: Become a part of our mission to protect and cherish the beauty of our planet. Together, let's create a future where every species thrives.



Zoo Management

Asset Management

- Insert Animal
- Update/View Animals
- Insert Building
- Insert Attraction
- Update/View Attractions
- Add employee
- Manage Employee
- Add animal species
- update/view species

Zoo Management

Daily Zoo Activity

- Attendance and Revenue

Reports

- Revenue report
- Species report
- Revenue
- five best days
- top three attraction
- Report on Revenue,total attendance

7. As you can see above there are 3 main categories:

- a. Asset Management
- b. Daily Zoo Activity
- c. Reports

8. Now if you want to add animal in the table, you can do it in 3 different ways:

- Add it from the webpage:

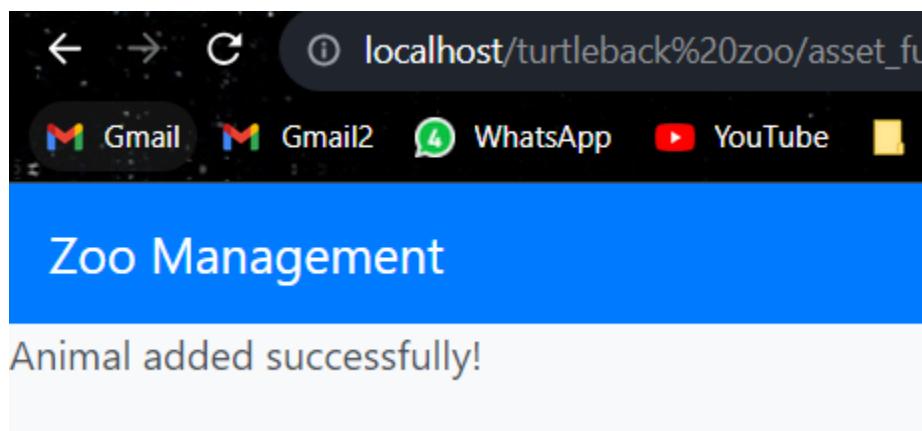
Animal Table

AnimalID	Species	LocationBuilding	LocationCage	BirthYear	CurrentStatus	Action
1	Lion	Building2	Cage1	2018	Healthy	<button>Update</button>
2	Elephant	Building2	Cage2	2010	Under Medical Care	<button>Update</button>
3	Giraffe	Building2	Cage3	2015	Maternal Leave	<button>Update</button>

The screenshot shows a web page titled "Add Animal" within a "Zoo Management" application. The page contains several input fields for entering animal information:

- Species: Crocodile
- Location Building: Building6
- Location Cage: Cage2
- Birth Year: 2005
- Current Status: Healthy

At the bottom left of the form is a blue "Add Animal" button.



- Add it from pHpMyAdmin:

I. Go to insert and type all the values to insert it:

The screenshot shows the 'Insert' page for the 'animals' table in the 'turtleback_zoo' database. The table structure is as follows:

AnimalID	Species	LocationBuilding	LocationCage	BirthYear	CurrentStatus
1	Lion	Building2	Cage1	2018	Healthy
2	Elephant	Building2	Cage2	2010	Under Medical Care
3	Giraffe	Building2	Cage3	2015	Maternal Leave
4	Crocodile	Building6	Cage2	2005	Healthy
5	Peacock	Building4	Cage5	1988	Unhealthy

II. Give an SQL command directly to insert any new values:

Go to SQL on top, type in your query and click go:

Server: 127.0.0.1 » Database: turtleback_zoo » Table: animals

Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers

Run SQL query/queries on table turtleback_zoo.animals:

```
1 INSERT INTO `animals`(`AnimalID`, `Species`, `LocationBuilding`, `LocationCage`, `BirthYear`, `CurrentStatus`) VALUES
(10,'Snake','Building8','Cage2',2017,'Healthy')
```

AnimalID	Species	LocationBuilding	LocationCage	BirthYear	CurrentStatus
1	Lion	Building2	Cage1	2018	Healthy
2	Elephant	Building2	Cage2	2010	Under Medical Care
3	Giraffe	Building2	Cage3	2015	Maternal Leave
4	Crocodile	Building6	Cage2	2005	Healthy
5	Peacock	Building4	Cage5	1988	Unhealthy
10	Snake	Building8	Cage2	2017	Healthy

SELECT * | SELECT | INSERT | UPDATE | DELETE | Clear | Format | Get auto-saved query | <>

Bind parameters

Bookmark this SQL query: []

Delimiter : | Show this query here again | Retain query box | Rollback when finished | Enable foreign key checks | Go |

9. All the other features can be seen below.

Source Codes Associated with the menu-driven application: (Attaching separate Zip file for all the codes)

1. Home Page of the Zoo Management:

```
C: > xampp > htdocs > Turtleback Zoo > index.php
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Turtleback Zoo Management System</title>
7
8      <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
9
10     <link rel="stylesheet" href="css/index.css">
11 </head>
12 <body>
13
14     <nav class="navbar navbar-expand-lg navbar-dark bg-primary">
15         <a class="navbar-brand" href="index.php">Zoo Management</a>
16         <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
17             <span class="navbar-toggler-icon"></span>
18         </button>
19         <div class="collapse navbar-collapse" id="navbarNav">
20             <ul class="navbar-nav">
21                 <li class="nav-item active">
22                     <a class="nav-link" href="index.php">Home</a>
23                 </li>
24                 <li class="nav-item">
25                     <a class="nav-link" href="asset_management.php">Asset Management</a>
26                 </li>
27                 <li class="nav-item">
28                     <a class="nav-link" href="daily_zoo_activity.php">Daily Zoo Activity</a>
29                 </li>
30             </ul>
31         </div>
32     </nav>
33
34     <section class="container section1 mt-4">
35         <h2>Welcome to Turtleback Zoo Management System</h2>
36         <p>
37             Unleash the power of the Turtleback Zoo Management System where efficiency meets excitement!
38             Dive into the heart of our operations, and experience a seamless blend of technology and wildlife
39             management.
40         </p>
41
42         <h3>Key Features:</h3>
43         <ul>
44             <li>Efficient Asset Management: Keep track of animals, buildings, attractions, and employees.</li>
45             <li>Daily Zoo Activity Monitoring: Record attendance, revenue, and other daily activities.</li>
46             <li>Comprehensive Management Reporting: Gain insights into the zoo's performance and make informed decisions.</li>
47         </ul>
48
49         <p>
50             <strong>Explore, Manage, Excel!</strong> Navigate through the system effortlessly using the menu above.
51             Your journey to a more organized and thriving zoo begins now!
52         </p>
53
54
55
56
57 </section>
```

```

C:/xampp/htdocs/turtlebackzoo/index.php
59     
60
61 </section>
62
63
64 <section class="container mt-4">
65     <h3>About Turtleback Zoo</h3>
66     <p>
67         Welcome to Turtleback Zoo - a realm of wonder and conservation! Our commitment to wildlife preservation
68         is as deep as our dedication to providing an unforgettable experience for visitors of all ages.
69     </p>
70
71     <p>
72         <strong>Embark on a Journey:</strong> From roaring lions to graceful dolphins, each exhibit tells a
73         story of nature's marvels. At Turtleback Zoo, we don't just showcase animals; we celebrate life and
74         inspire a passion for the wild.
75     </p>
76
77     <p>
78         <strong>Join the Adventure:</strong> Become a part of our mission to protect and cherish the beauty
79         of our planet. Together, let's create a future where every species thrives.
80     </p>
81
82     
83
84 </section>
85
86 <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
87 <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.1/dist/umd/popper.min.js"></script>

```

```

69     </p>
70
71     <p>
72         <strong>Embark on a Journey:</strong> From roaring lions to graceful dolphins, each exhibit tells a
73         story of nature's marvels. At Turtleback Zoo, we don't just showcase animals; we celebrate life and
74         inspire a passion for the wild.
75     </p>
76
77     <p>
78         <strong>Join the Adventure:</strong> Become a part of our mission to protect and cherish the beauty
79         of our planet. Together, let's create a future where every species thrives.
80     </p>
81
82     
83
84 </section>
85
86 <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
87 <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.1/dist/umd/popper.min.js"></script>
88 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
89 </body>
90 </html>
91

```

2. The TurtleZoo Management System has menus that has various functionalities:

Asset Management:

```

asset_management.php
C: > xampp > htdocs > Turtleback Zoo > asset_management.php
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Turtleback Zoo Management System</title>
7
8      <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
9      <link rel="stylesheet" href="css/assetman.css">
10
11 </body>
12
13 <nav class="navbar navbar-expand-lg navbar-dark bg-primary">
14     <a class="navbar-brand" href="index.php">Zoo Management</a>
15 </nav>
16
17     <div class="container">
18         <h2>Asset Management</h2>
19         <ul>
20             <li><a href="asset_functions.php?action=insert_animal">Insert Animal</a></li>
21             <li><a href="update_animal.php">Update/View Animals</a></li>
22             <li><a href="asset_functions.php?action=insert_building">Insert Building</a></li>
23             <li><a href="asset_functions.php?action=insert_attraction">Insert Attraction</a></li>
24             <li><a href="attraction_update.php">Update/View Attractions</a></li>
25             <li><a href="add_employees.php">Add employee</a></li>
26             <li><a href="employee_update.php">Manage Employee</a></li>
27             <li><a href="species.php">Add animal species</a></li>
28             <li><a href="modify.php">update/view species</a></li>
29
30         </ul>
31     </div>
32     <?php include 'footer.php'; ?>
33
34

```

Daily Zoo Activity:

```

daily_activity.php
1  <?php include 'header.php'; ?>
2
3  <h2>Daily Zoo Activity</h2>
4  <ul>
5      <li><a href="daily_activity_functions.php?action=attendance_and_revenue">Attendance and Revenue</a></li>
6
7  </ul>
8
9
10 <?php include 'footer.php'; ?>
11

```

Reports:

```

C:\xampp\htdocs\Turtleback Zoo\reports.php
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>Turtleback Zoo Management System</title>
7     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
8     <link rel="stylesheet" href="css/assetsman.css">
9   </head>
10  <body>
11    <nav class="navbar navbar-expand-lg navbar-dark bg-primary">
12      <a class="navbar-brand" href="index.php">Zoo Management</a>
13    </nav>
14    <div class="container">
15      <h2>Reports</h2>
16      <ul>
17        <li><a href="report_revenue.php">Revenue report</a></li>
18        <li><a href="report_species.php">Species report</a></li>
19        <li><a href="reports/revenue_report.php">Revenue</a></li>
20        <li><a href="reports/best_months.php">Five best days</a></li>
21        <li><a href="reports/top_three_attraction.php">Top three attraction</a></li>
22        <li><a href="reports/query.php">Report on Revenue, total attendance</a></li>
23      </ul>
24    </div>
25    <?php include 'footer.php'; ?>
26  </body>
27  </html>
28
29
30

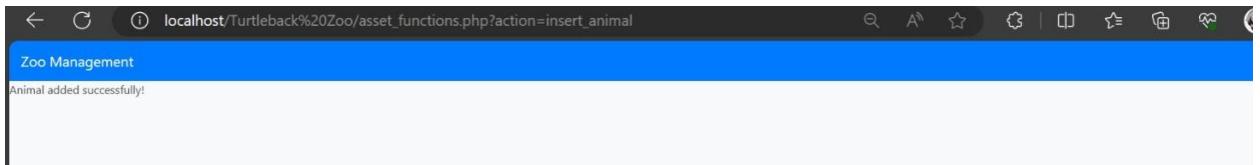
```

Insert/Update/View:

Here in the application under Asset Management, we are inserting the Animal Details

Species:	Cheetah
Location Building:	build 678
Location Cage:	cage 567
Birth Year:	1956
Current Status:	Healthy
<input type="button" value="Add Animal"/>	

Successful execution of Inserting Animal



The insertion of Animal gets added in the XAMPP PHP Admin as well.

	Employee ID	Name	Building	Cage	Birth Year	Status
	23	flft	rgtfl	ghfyt	456	healthy
	24	Cheetah	build 678	cage 567	1956	Healthy

Here we have added the details of the Animal Cheetah this got added in the PHP Admin in the above fig.

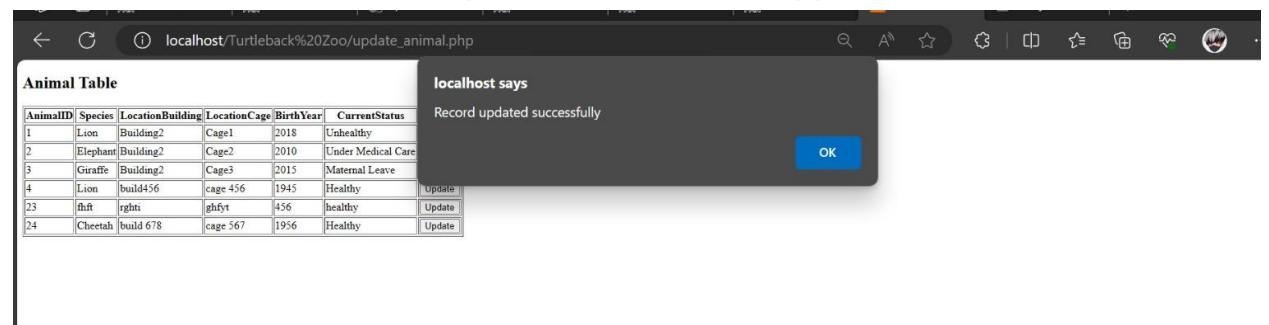
Update/view Animal:



AnimalID	Species	LocationBuilding	LocationCage	BirthYear	CurrentStatus	Action
1	Lion	Building2	Cage1	2018	Healthy	<input type="button" value="Update"/>
2	Elephant	Building2	Cage2	2010	Under Medical Care	<input type="button" value="Update"/>
3	Giraffe	Building2	Cage3	2015	Maternal Leave	<input type="button" value="Update"/>
4	Lion	build456	cage 456	1945	Healthy	<input type="button" value="Update"/>
23	flft	rgfnt	ghfyt	456	healthy	<input type="button" value="Update"/>
24	Cheetah	build 678	cage 567	1956	Healthy	<input type="button" value="Update"/>

In the above fig, the details of the Animal are present. Now we will perform the Update operation.

Successful execution of the Update Animal in the Application



In the application when we observe we have updated Lion Current Status as Unhealthy, similarly in the XAMPP PHP Admin the Lion's current status is updated as shown below



Buildings:

Similarly in the Menu-driven application and XAMPP PHP Admin the Building information gets inserted and viewed. The following figs is as follows:

Zoo Management

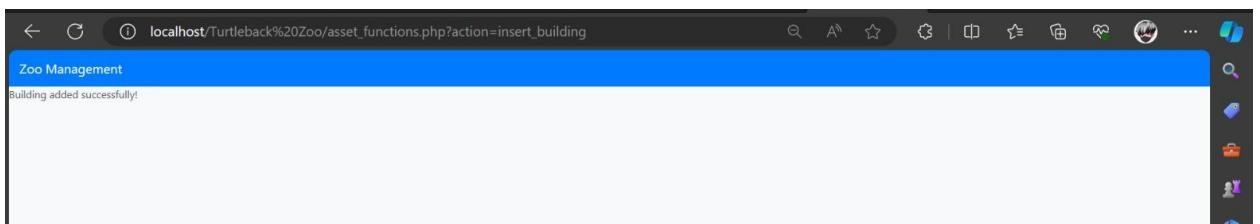
Add Building

Name: Jersey City

Purpose: Admission

Floors: 4

Square Footage: 5



In the XAMPP the following information is inserted and viewed.

	BuildingID	Name	Purpose	Floors	SquareFootage
<input type="checkbox"/>	1	Building1	Animal Shelter	2	2000
<input type="checkbox"/>	2	Building2	Attraction	3	3000
<input type="checkbox"/>	3	Building3	Maintenance	1	1000
<input type="checkbox"/>	4	Jersey City	Admission	4	5

In the same way, adding the Attractions and updating/viewing the attractions:

Zoo Management

Add Attraction

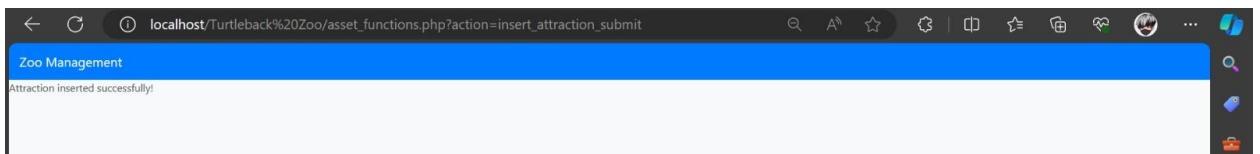
Attraction Name: Atlantic City

Building ID: 234

Shows Per Day: 58

Ticket Price: 300

Successful execution of the Attraction added:



It is reflecting in the XAMPP PHP Admin:

1	Monkey Show	2	3	10.00	Update
2	Big Cat Show	2	2	12.50	Update
3	Aquatic Show	2	4	15.00	Update
23	gar	56	3	34.00	Update
45	ghti	67	89	34.00	Update
56	Mammals	45	34	23.00	Update
57	Atlantic City	234	58	300.00	Update

Successful execution of the Record Updation

localhost says
Record updated successfully

AttractionID	Name	BuildingID	ShowsPerDay	TicketPrice	Action
1	Monkey Show	2	3	10.00	Update
2	Big Cat Show	2	2	12.50	Update
3	Aquatic Show	2	4	15.00	Update
23	gar	56	3	34.00	Update
45	ghti	67	89	34.00	Update
56	Mammals	45	34	23.00	Update
57	Atlantic City	234	58	300.00	Update

The record got updated in the XAMPP PHP Admin:

Showing rows 0 - 6 (7 total, Query took 0.0003 seconds.)

SELECT * FROM `attractions`

AttractionID	Name	BuildingID	ShowsPerDay	TicketPrice
1	Monkey Show	2	3	10.00
2	Big Cat Show	2	2	12.50
3	Aquatic Show	2	4	15.00
23	gar	56	3	34.00
45	ghti	67	89	34.00
56	Mammals	45	34	23.00
57	Atlantic City	234	58	300.00

Inserting and Updating Employees:

Add Employee

Name:

Address:

Start Date:

Employee Type:

Hourly Rate:

[Add Employee](#)

Got updated in the PHP Admin:

The screenshot shows the phpMyAdmin interface for the 'employees' table in the 'turtleback_zoo' database. The table has columns: EmployeeID, Name, Address, StartDate, EmployeeType, and HourlyRate. The data is as follows:

EmployeeID	Name	Address	StartDate	EmployeeType	HourlyRate
1	John Doe	123 Main St	2023-01-01	Maintenance	15.00
2	Jane Smith	456 Oak St	2023-01-15	Animal Care Specialist	18.00
3	Bob Johnson	789 Pine St	2023-02-01	Customer Service	12.50
4	Jam	465 meadow road	2023-12-05	Permanent	45.00
5	Joy	205 SKEM	2023-12-06	Veterinarian	45.00

Hourly Wages is also included in the Employee insertion and updation itself in the above figs.

Management Reports:

Revenue Report for all Shows:

The screenshot shows a web-based report titled "Zoo financial Reporting" under "Revenue Report for All Shows". The table displays attraction details and their total revenue:

AttractionID	Attraction Name	Date	Tickets Sold	Revenue Earned
1	Monkey Show	2023-11-01	50	\$500.00
2	Big Cat Show	2023-11-01	30	\$375.00
3	Aquatic Show	2023-11-01	40	\$600.00
23	gar	2023-12-12	2000	\$1000.00

Total Revenue: \$2475

Top 5 days Revenue Report:

The screenshot shows a report titled "Top 5 Revenue Days Report" listing dates and their corresponding revenue earned:

Date	Revenue Earned
2023-10-03	3000.00
2023-11-28	2500.00
2023-12-12	1000.00
2023-11-01	600.00
2023-11-01	500.00

Top 3 Attractions:

The screenshot shows a report titled "Top 3 Attractions by Revenue" listing attraction IDs and names along with their total revenue:

Attraction ID	Name	Total Revenue
2	Big Cat Show	3375.00
23	gar	1000.00
3	Aquatic Show	600.00

Average Revenue Generated:

The screenshot shows a report titled "Compute Average Revenue" with input fields for 'Begin Date' (05-12-2023) and 'End Date' (08-12-2023), and a button to "Compute Average Revenue". The resulting table shows the average revenue for different item types:

Item Type	Item ID	Average Revenue
Attraction	1	500.000000
Attraction	2	375.000000
Attraction	3	600.000000

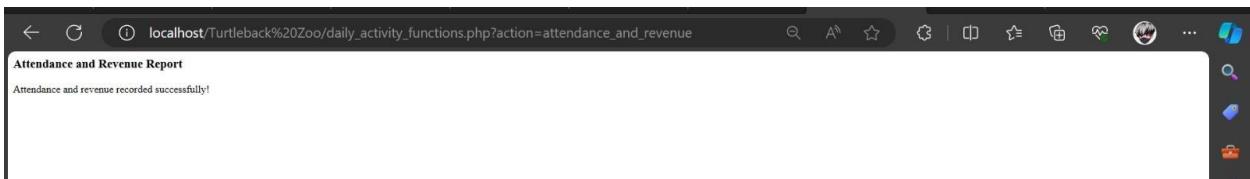
Daily Zoo Activity:
Attendance and Revenue Report details are listed below:



Attendance and Revenue Report

Attraction ID: 6 Date: 05-12-2023 Tickets Sold: 90 Revenue Earned: 300 Record Attendance and Revenue

When Recorded successfully:



Attendance and Revenue Report

Attendance and revenue recorded successfully!