

DBMS

Mini-project Report

“Tour Management System”

Submitted By: Group-7

Name	Roll No	PRN
Amanjot Singh Saini	10	1032221627
Shivani Samdani	13	1032221681
Sai Kale	23	1032222245
Shriya Ratna	32	1032222335

Under the Guidance
of Faculty Name

Prof. Pradnya S. Kulkarni

AT



Dr. Vishwanath Karad

**MIT WORLD PEACE
UNIVERSITY** | PUNE

TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

School of Computer Engineering and Technology

Acknowledgements

We would like to express our deep gratitude and sincere thanks to our DBMS Professor Prof. Pradnya S. Kulkarni for their constant valuable advice and constructive inputs which enhanced the quality of our work. This project could not have been completed successfully without their help.

Lastly, we would also like to thank our parents for their love and guidance without which, nothing can be accomplished.

Abstract Idea

A Tour Management System is a web-based application designed to streamline and automate various travel-related processes for a customized user experience. Using HTML, CSS, (alongside Bootstrap, jQuery), PHP, POD and MySQL, a Tour Management System can provide a comprehensive platform for travel agencies or companies to manage their operations efficiently.

This system would allow users to search for the desired destination (countries), skim through various available deals with different no. of days, hotel included, flight tickets, daily places(cities/ itinerary) and book a trip according to their convenience, providing them with real-time availability and pricing information. Users can access their travel details, view invoices, and make payments securely through the system.

Furthermore, this system also contains a user login/registration and admin login handle. It can integrate with external APIs for fetching data from third-party providers, ensuring access to the latest information.

With HTML and CSS, the system's user interface can be developed to offer an intuitive and visually appealing experience. PHP would handle the server-side scripting, enabling dynamic functionality and database interactions using MySQL for data storage and retrieval.

In summary, a Tour Management System using HTML, CSS, Bootstrap, PHP, and MySQL would provide a robust and user-friendly platform for efficient travel planning, booking, and management.

INDEX

Sr. No.	Content	Page No.
1	Introduction	5
2	Problem Statement	7
3	System Flow Architecture with Database Design (ER Diagram)	8
4	GUI (Screenshots) with Client side validations	11
5	Server-side database handling details	18
6	Code Snippets	22
7	Conclusions	26
8	Appendix	27

INTRODUCTION

The travel industry is highly dynamic and demands efficient management of various travel-related processes. To address this need, a Tour Management System (TMS) powered by HTML, CSS, Bootstrap, PHP, and MySQL can revolutionize how travel agencies and companies operate. This report provides an overview of the benefits and functionalities of a TMS, highlighting its potential to streamline travel operations.

This mini project, developed as a part of a Database Management System(DBMS) course, aims to provide an efficient solution for organizing and tracking various aspects of tour management. The system is designed to cater to the needs of travel agencies, tour operators, or any entity involved in planning and executing tours.

KEY FEATURES:

1. **User Authentication and Authorization:** Secure login functionality for administrators, staff, and customers. Also, different levels of access control to ensure data security.
2. **Customer management:** Maintain customer profiles with details such as name, contact information, and booking history. Enable customers to register, log in, and view their booking status.
3. **Destination Management:** Store information about various tour destinations, including details on attractions, accommodations, and travel arrangements.

4. **Booking and Reservations:** Allow customers to browse available tour packages and make bookings. Provide a reservation system to manage seat availability and confirmations.
5. **Deals Section:** Country-based deals section enabling users to filter by price range, facilitating seamless searches for desired offers. Simplify browsing, select countries, set price filters, and discover exclusive deals effortlessly.

This Tour Management System mini project provides a comprehensive solution for managing the complexities of tour operations. By leveraging a robust database management system and web technologies, the system ensures efficiency, accuracy, and a seamless experience for both tour operators and customers. The project allows students to gain hands-on experience in designing and implementing a practical database system while addressing real-world challenges in the tour management domain.

PROBLEM STATEMENT

In the contemporary travel and tourism industry, the efficient management of tours and related activities poses a significant challenge. Tour operators and travel agencies often grapple with manual and error-prone processes, resulting in inefficiencies, customer dissatisfaction, and a lack of centralized information. To address these challenges, the need for an integrated Tour Management System is evident. This mini project aims to develop a Database Management System(DBMS) solution that streamlines and automates the tour management process, providing a centralized platform for efficient data management and user interaction.

System Flow Architecture with Database Design (ER diagram)

Entities:

1. Admin:

Attributes: id(primary key), adminname, email, mypassword, created_at

2. Users:

Attributes: id(primary key), email, username, mypassword, created_at

3. Countries:

Attributes: id(primary key), name, image, continent, population, territory, description, created_at

4. Cities:

Attributes: id(primary key), name, image, trip_days, price, country_id(foreign key references Countries), created_at

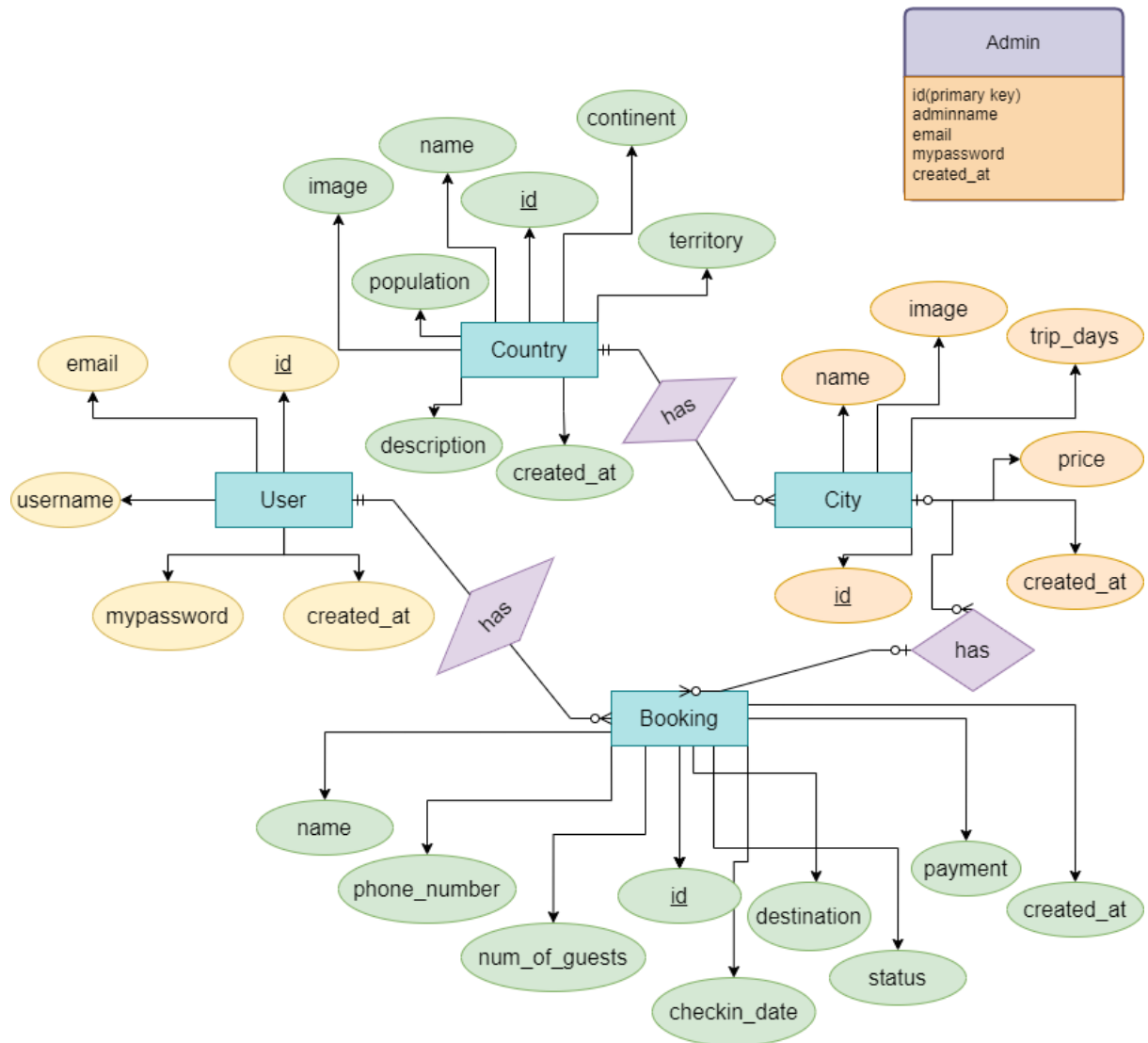
5. Bookings:

Attributes: id(primary key), name, phone_number, num_of_guests, checkin_date, destination, status, city_id(foreign key references Cities), uder_id(foreign key references User), payment, created_at

Relationships:

1. Countries-Cities (one to many)
 - A country has many cities that can be visited.
 - Foreign key: country_id in Cities entity
2. User-Booking(one to many)
 - A user can have multiple bookings
 - Foreign key: user_id in Bookings entity
3. Cities-Booking(one to many)
 - There can be multiple bookings for the same city.
 - Foreign key: city_id in Bookings entity

ER-Diagram

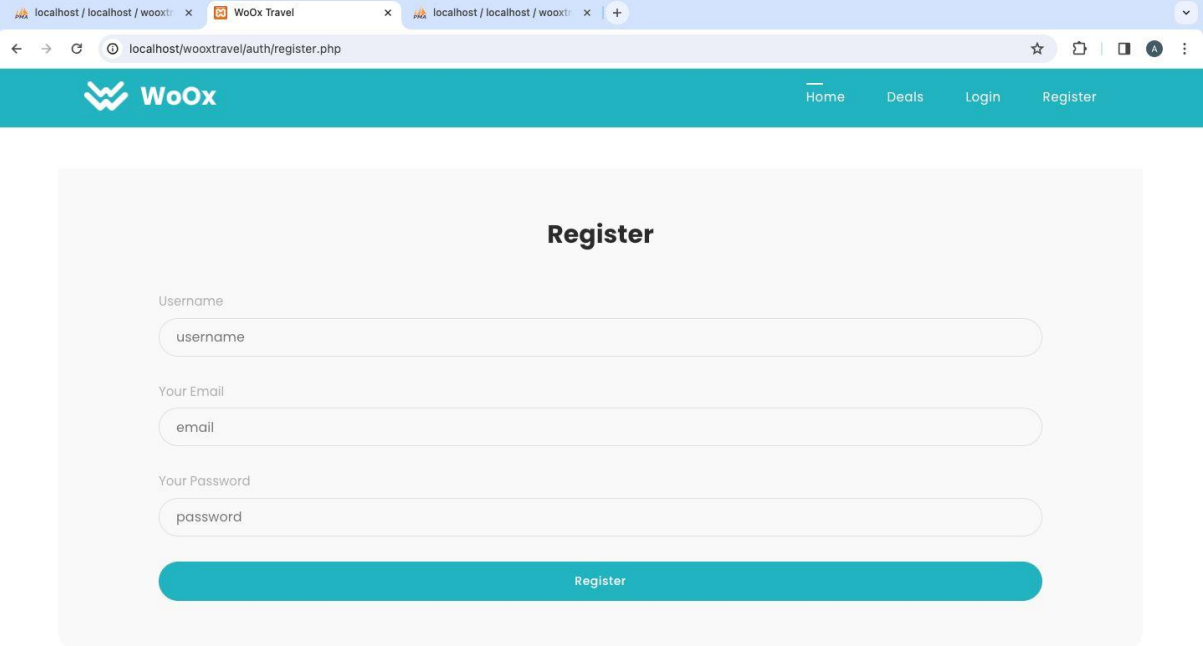


Link for ER-Diagram-

https://drive.google.com/file/d/1n5Sz_WtVBbS5idOCpMzWK77ozVRLaek1/view?usp=sharing

GUI (Screenshots) with Client side validations

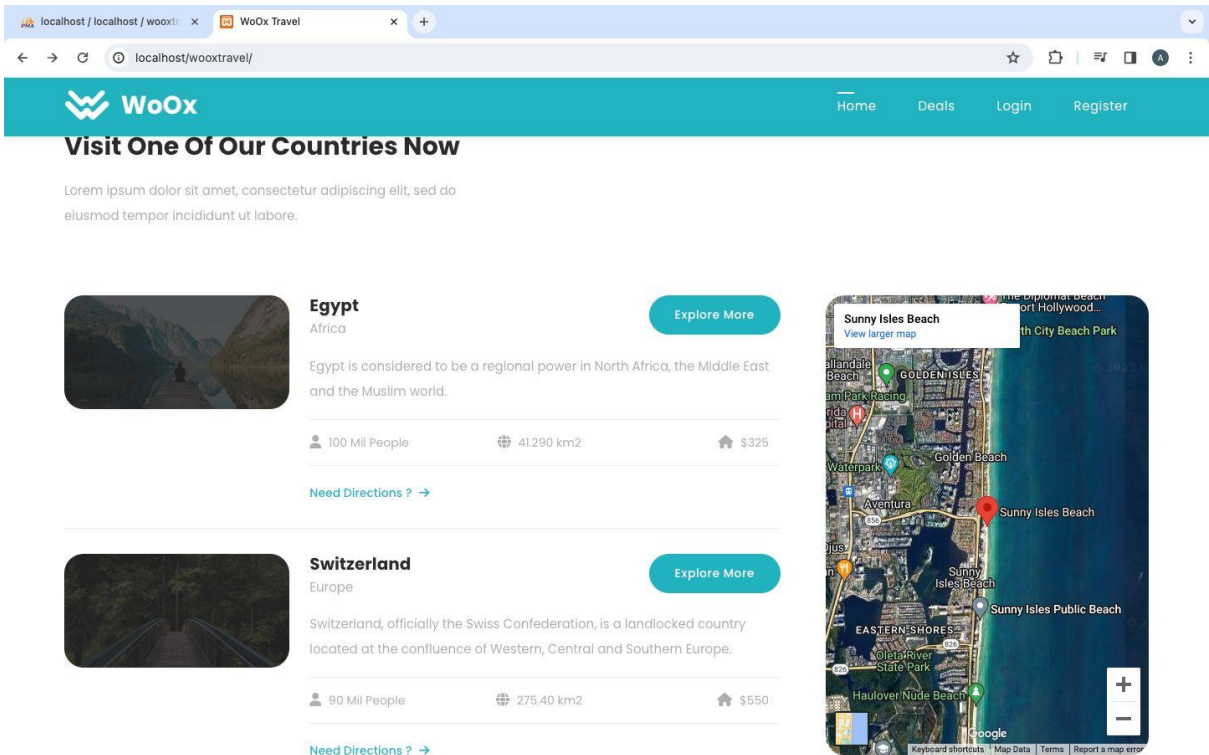
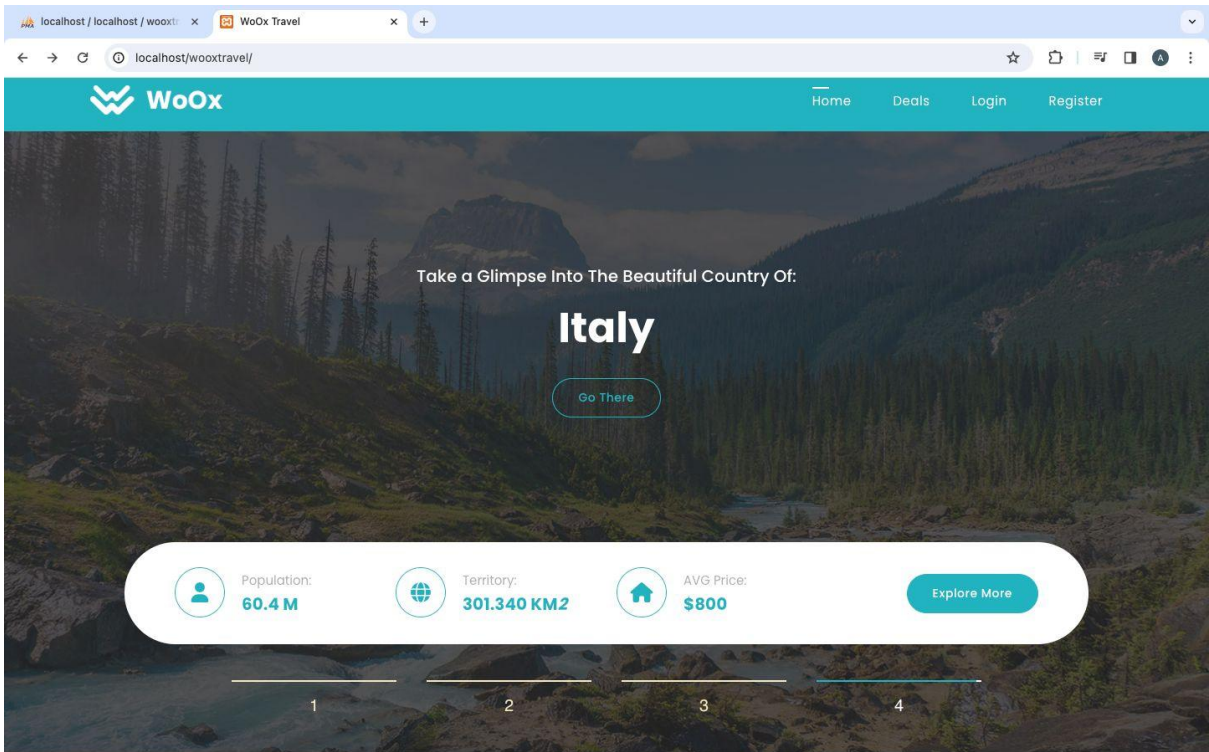
1-User Login/Registration Page



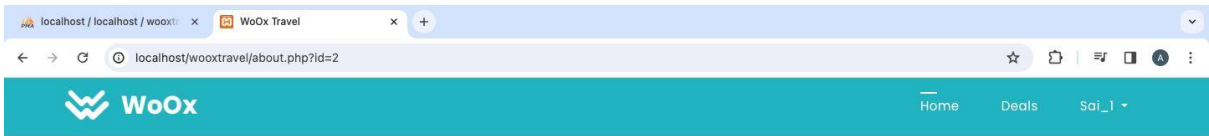
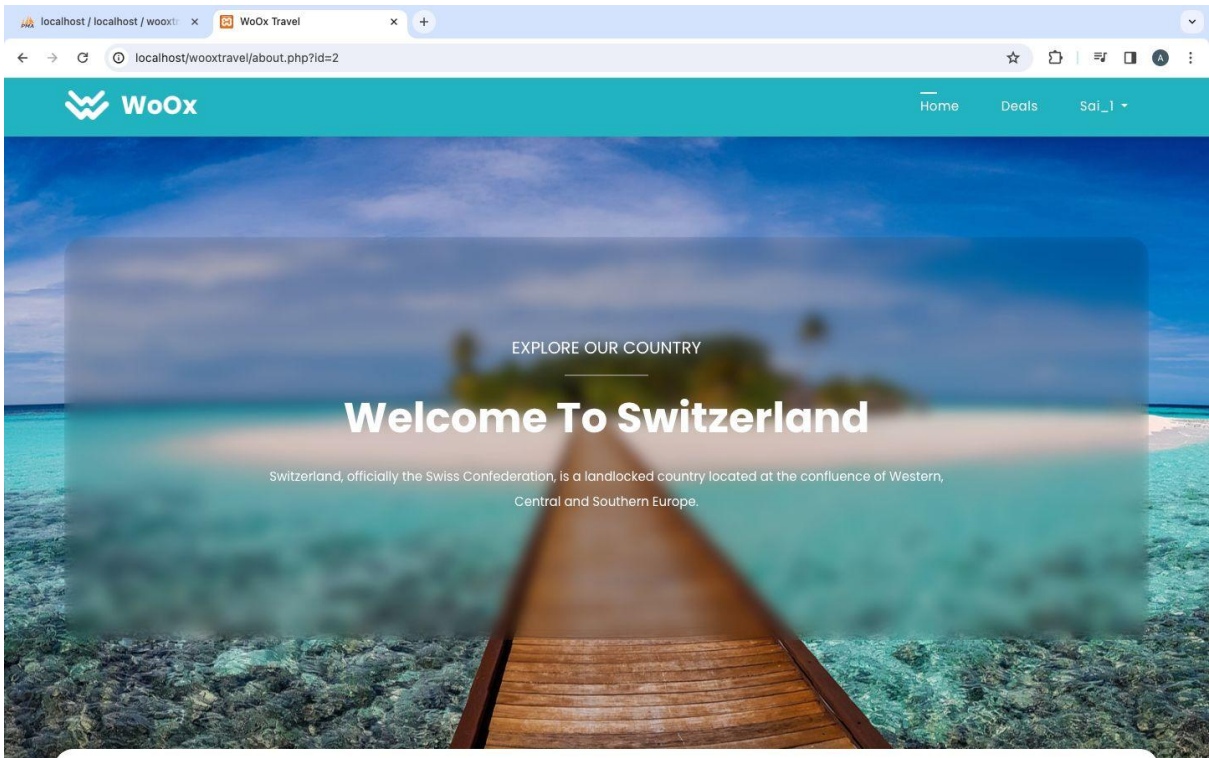
The screenshot shows a web browser window with the URL `localhost/wooxtravel/auth/register.php`. The page features a teal header with the WoOx logo and navigation links: Home, Deals, Login, and Register. The main content area is a light gray box titled "Register" containing three input fields: "Username" (with placeholder "username"), "Your Email" (with placeholder "email"), and "Your Password" (with placeholder "password"). A teal "Register" button is positioned at the bottom of the form.

This page will load when the user who isn't registered would want to register.

2-Home Page

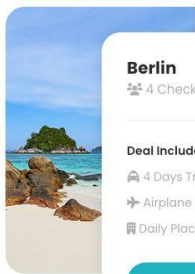


3-About Us Page



Best Weekly Offers In Each City

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore.



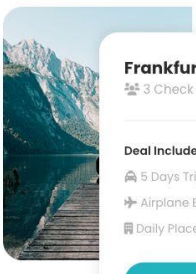
Berlin \$600

4 Check Ins /person

Deal Includes:

- 4 Days Trip > Hotel Included
- Airplane Bill Included
- Daily Places Visit

Make A Reservation



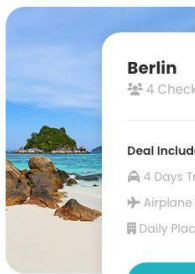
Frankfurt \$500

3 Check Ins /person

Deal Includes:

- 5 Days Trip > Hotel Included
- Airplane Bill Included
- Daily Places Visit

Make A Reservation



Berlin \$600

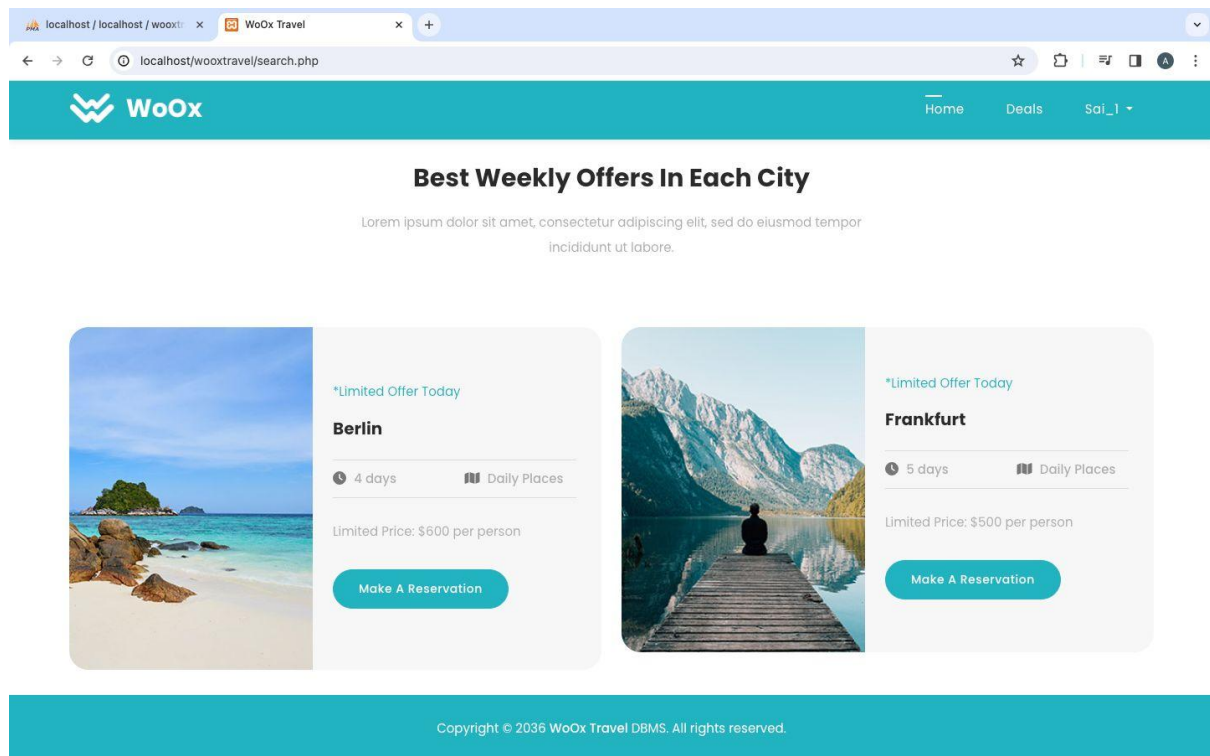
4 Check Ins /person

Deal Includes:

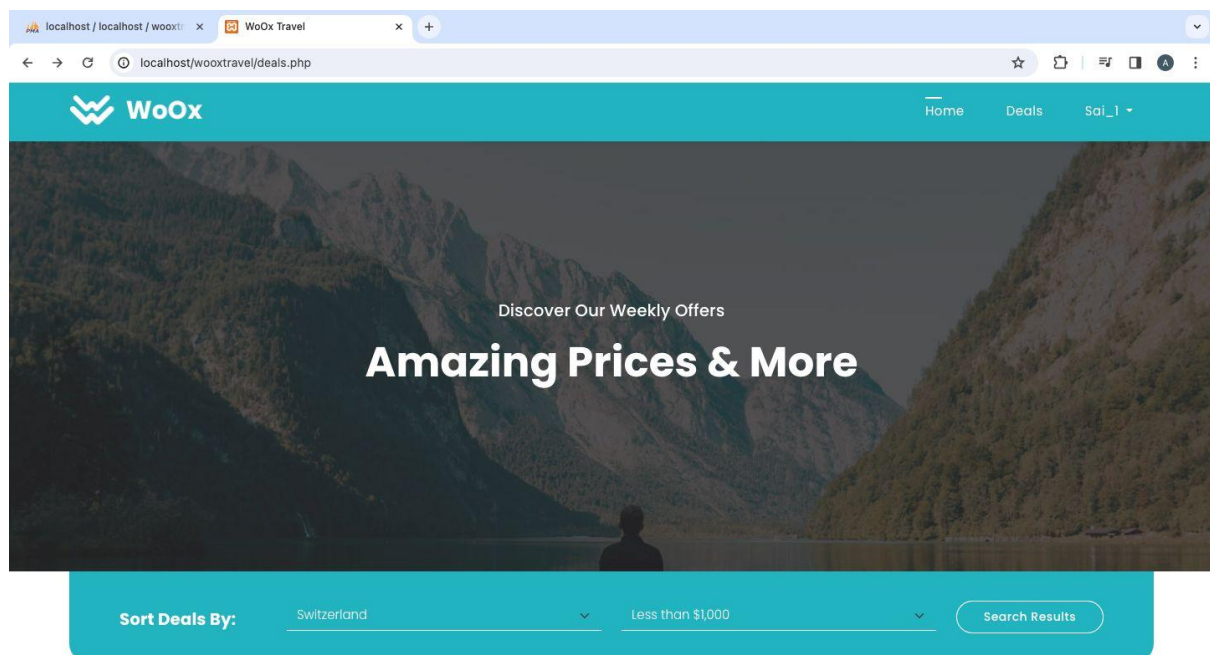
- 4 Days Trip > Hotel Included
- Airplane Bill Included
- Daily Places Visit

Make A Reservation

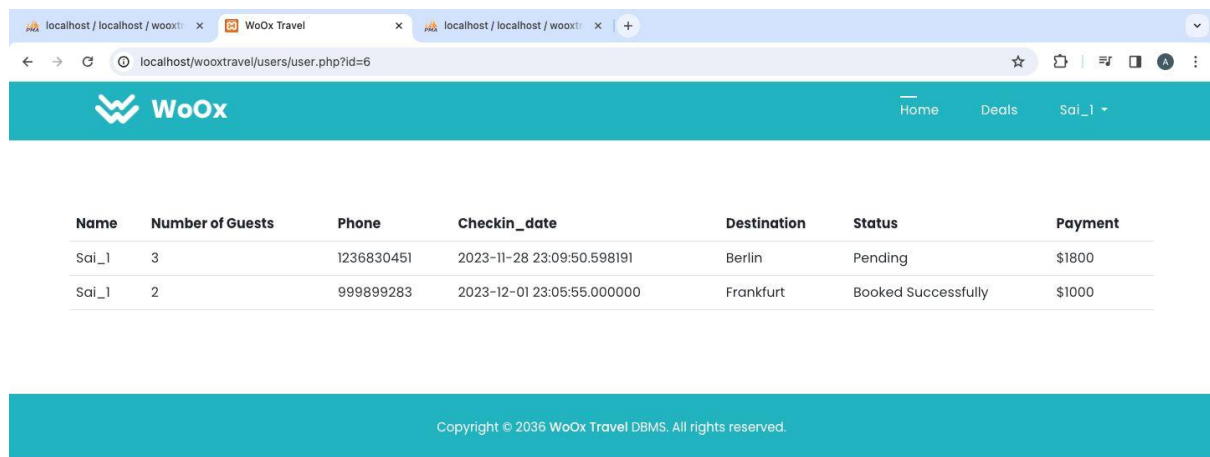
4-Result after search



5- 'Search Deals' page



6-User's Booking status

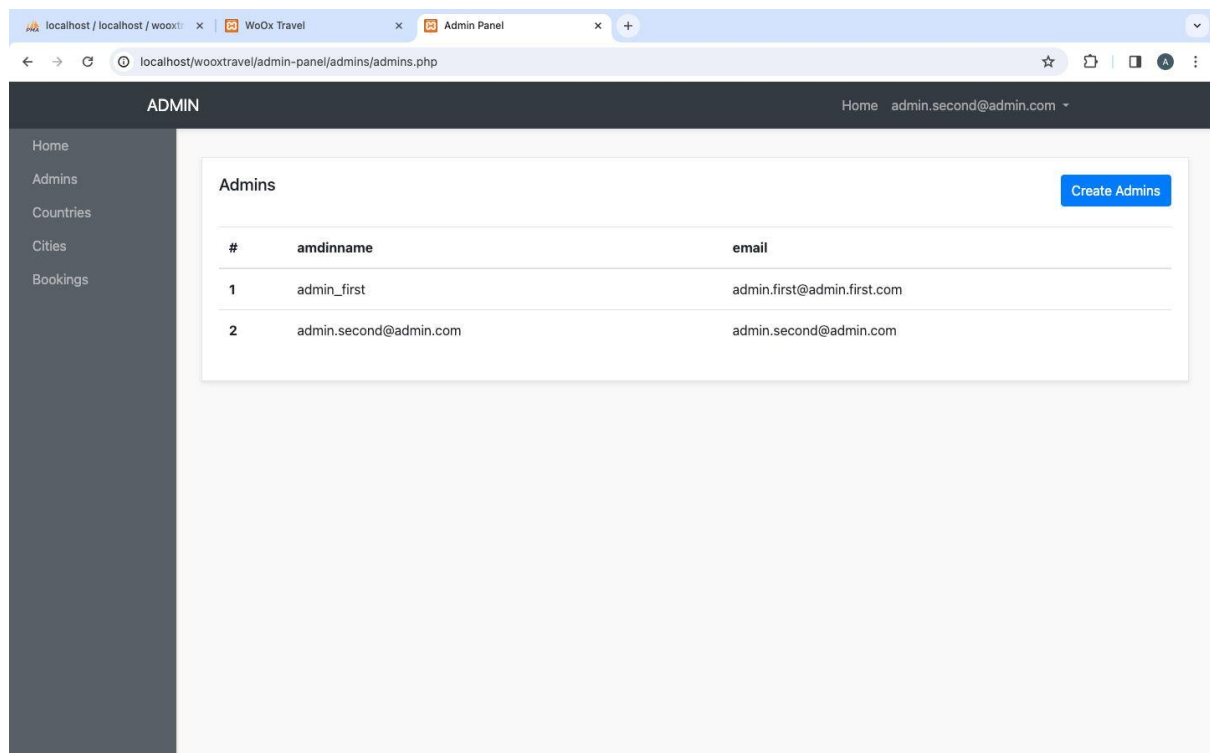


Name	Number of Guests	Phone	Checkin_date	Destination	Status	Payment
Sai_1	3	1236830451	2023-11-28 23:09:50.598191	Berlin	Pending	\$1800
Sai_1	2	999899283	2023-12-01 23:05:55.000000	Frankfurt	Booked Successfully	\$1000

Copyright © 2036 WoOx Travel DBMS. All rights reserved.

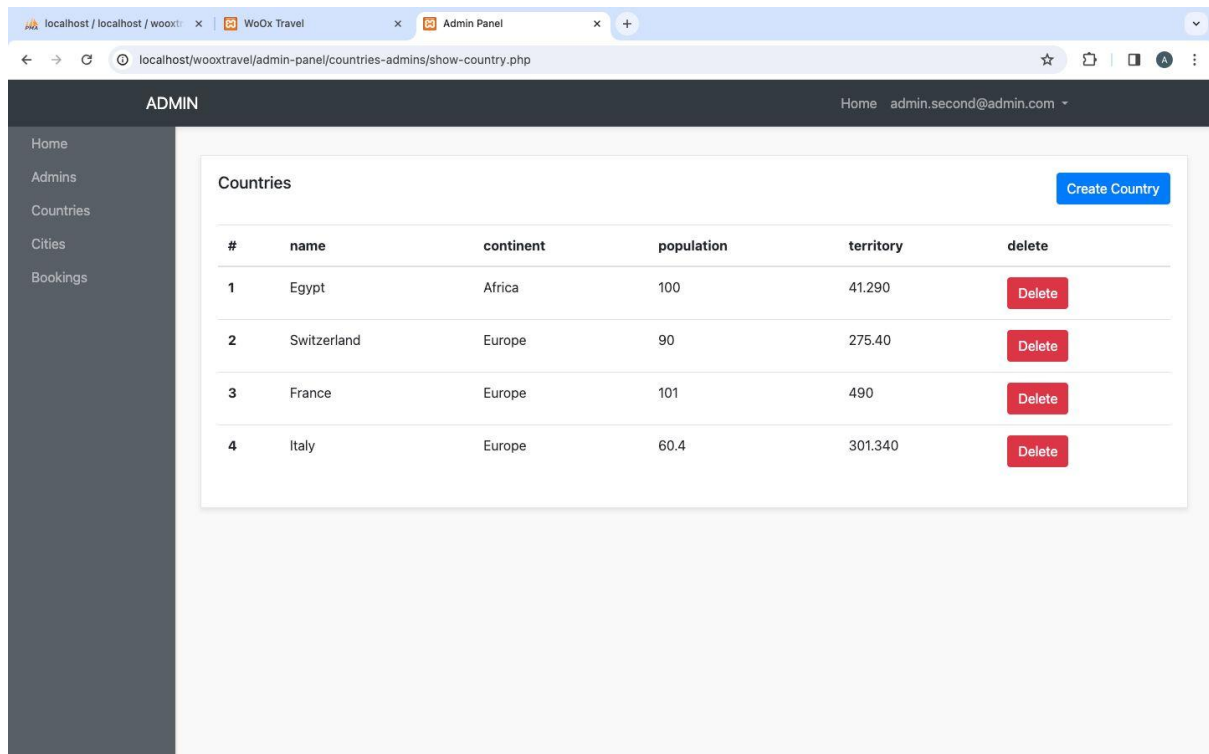
ADMIN SIDE-

1-Admin Page



#	amdinname	email
1	admin_first	admin.first@admin.first.com
2	admin.second@admin.com	admin.second@admin.com

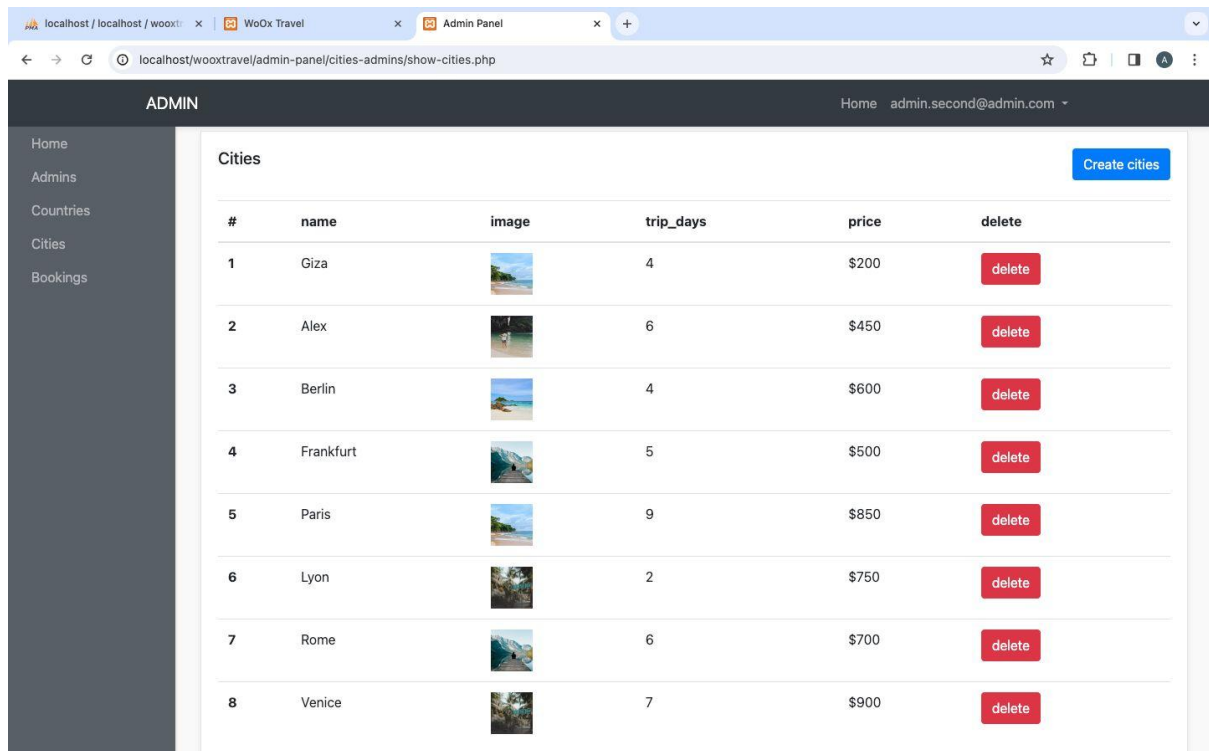
2-Countries Admin Page











The screenshot shows the '2-Countries Admin Page' in a web browser. The browser's address bar displays the URL `localhost/wooxtravel/admin-panel/countries-admins/show-country.php`. The page features a dark sidebar on the left with navigation links: Home, Admins, Countries, Cities, and Bookings. The main content area is titled 'ADMIN' and includes a user profile 'admin.second@admin.com'. Below this, a 'Countries' section contains a table with columns: #, name, continent, population, territory, and delete. A 'Create Country' button is located at the top right of the table. The table lists four countries: Egypt (Africa, population 100, territory 41.290), Switzerland (Europe, population 90, territory 275.40), France (Europe, population 101, territory 490), and Italy (Europe, population 60.4, territory 301.340). Each row has a red 'Delete' button.

#	name	continent	population	territory	delete
1	Egypt	Africa	100	41.290	Delete
2	Switzerland	Europe	90	275.40	Delete
3	France	Europe	101	490	Delete
4	Italy	Europe	60.4	301.340	Delete

3-Cities Admin Page



The screenshot shows the '3-Cities Admin Page' in a web browser. The browser's address bar displays the URL `localhost/wooxtravel/admin-panel/cities-admins/show-cities.php`. The page features a dark sidebar on the left with navigation links: Home, Admins, Countries, Cities, and Bookings. The main content area is titled 'ADMIN' and includes a user profile 'admin.second@admin.com'. Below this, a 'Cities' section contains a table with columns: #, name, image, trip_days, price, and delete. A 'Create cities' button is located at the top right of the table. The table lists eight cities: Giza (4 trip days, \$200), Alex (6 trip days, \$450), Berlin (4 trip days, \$600), Frankfurt (5 trip days, \$500), Paris (9 trip days, \$850), Lyon (2 trip days, \$750), Rome (6 trip days, \$700), and Venice (7 trip days, \$900). Each row has a red 'delete' button.

#	name	image	trip_days	price	delete
1	Giza		4	\$200	delete
2	Alex		6	\$450	delete
3	Berlin		4	\$600	delete
4	Frankfurt		5	\$500	delete
5	Paris		9	\$850	delete
6	Lyon		2	\$750	delete
7	Rome		6	\$700	delete
8	Venice		7	\$900	delete

4-Bookings Admin Page

The screenshot displays the 'ADMIN' interface of the 'WoOx Travel' system. The left sidebar contains navigation links: Home, Admins, Countries, Cities, and Bookings. The main content area is titled 'Bookings' and features a table with the following data:

#	name	phone_number	num_of_geusts	checkin_date	destination	payment	status
2	John Doe	2929333	2	2023-11-28 11:22:43.598502	Frankfurt	\$	Booked Successfully
6	aman@gmail.com	999899283	3	2023-11-30 11:07:15.000000	Berlin	\$1000	Booked Successfully
8	Sai_1	1236830451	3	2023-11-28 23:09:50.598191	Berlin	\$1800	Pending
19	Sai_1	999899283	2	2023-12-01 23:05:55.000000	Frankfurt	\$1000	Booked Successfully

Server-side database handling details

1-Admins

The screenshot shows the phpMyAdmin interface for the 'admins' table. The table structure is as follows:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(4)			No	None		AUTO_INCREMENT	Change Drop More
2	adminname	varchar(200)	utf8mb4_general_ci		No	None			Change Drop More
3	email	varchar(200)	utf8mb4_general_ci		No	None			Change Drop More
4	mypassword	varchar(200)	utf8mb4_general_ci		No	None			Change Drop More
5	created_at	timestamp			No	current_timestamp()			Change Drop More

The interface also shows the 'Indexes' section with a primary index on the 'id' column.

2-Users

The screenshot shows the phpMyAdmin interface for the 'users' table. The table structure is as follows:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(3)			No	None		AUTO_INCREMENT	Change Drop More
2	email	varchar(200)	utf8mb4_general_ci		No	None			Change Drop More
3	username	varchar(200)	utf8mb4_general_ci		No	None			Change Drop More
4	mypassword	varchar(200)	utf8mb4_general_ci		No	None			Change Drop More
5	created_at	timestamp			No	current_timestamp()			Change Drop More

The interface also shows the 'Indexes' section with a primary index on the 'id' column.

3-Countries

localhost/phpmyadmin/index.php?route=/table/structure&db=wooxtravel&table=countries

Server: localhost - Database: wooxtravel - Table: countries

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(3)			No	None		AUTO_INCREMENT	Change Drop More
2	name	varchar(200)	utf8mb4_general_ci		No	None			Change Drop More
3	image	varchar(200)	utf8mb4_general_ci		No	None			Change Drop More
4	continent	varchar(200)	utf8mb4_general_ci		No	None			Change Drop More
5	population	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
6	territory	varchar(30)	utf8mb4_general_ci		No	None			Change Drop More
7	description	text	utf8mb4_general_ci		No	None			Change Drop More
8	created_at	timestamp			No	current_timestamp()			Change Drop More

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY		BTREE	Yes	No	id	2	A	No	

Partitions

No partitioning defined!

4-Cities

localhost/phpmyadmin/index.php?route=/table/structure&db=wooxtravel&table=cities

Server: localhost - Database: wooxtravel - Table: cities

Table structure

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(3)			No	None		AUTO_INCREMENT	Change Drop More
2	name	varchar(200)	utf8mb4_general_ci		No	None			Change Drop More
3	image	varchar(200)	utf8mb4_general_ci		No	None			Change Drop More
4	trip_days	int(4)			No	None			Change Drop More
5	price	varchar(4)	utf8mb4_general_ci		No	None			Change Drop More
6	country_id	int(4)			No	None			Change Drop More
7	created_at	timestamp			No	current_timestamp()			Change Drop More

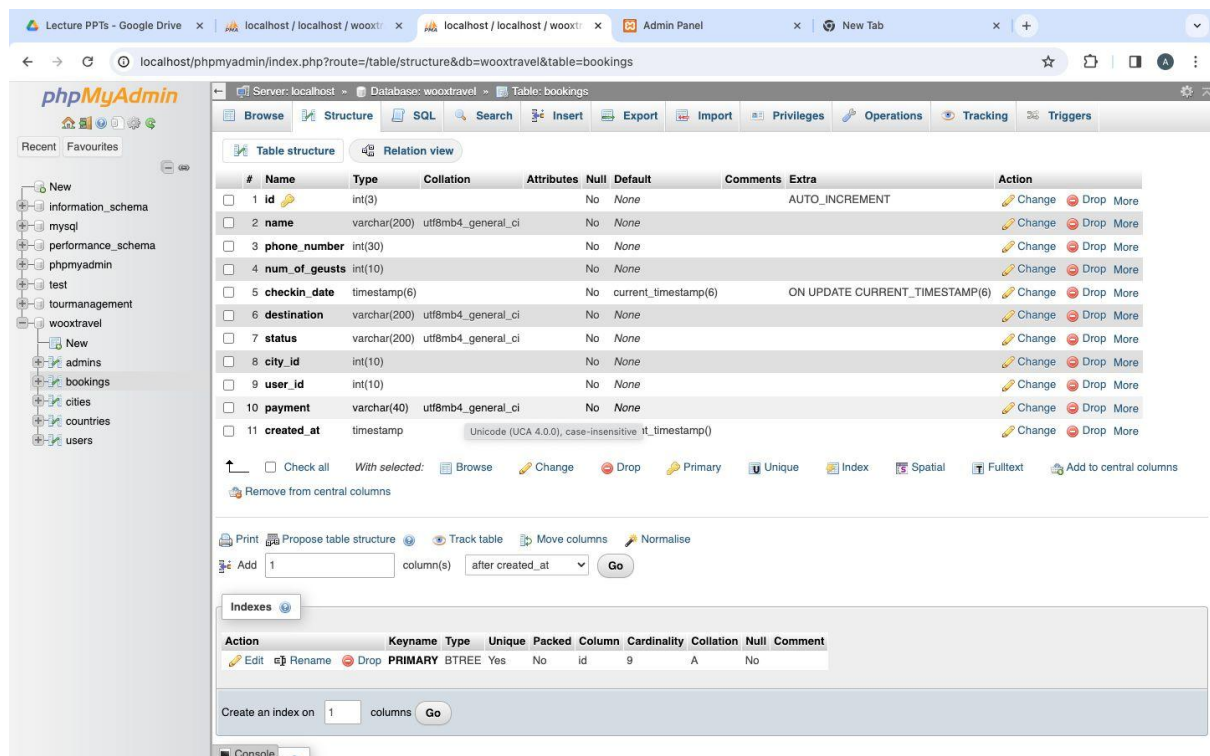
Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
PRIMARY		BTREE	Yes	No	id	4	A	No	

Partitions

No partitioning defined!

5-Bookings



1. Database Management System (DBMS): -

Choose a suitable database platform. In this case we have used MySQL. - Install and configure the DBMS on your server to create and manage the database. 18

2. Database Design: -

Design the database schema to represent the entities, relationships, and attributes required for the project. - Identify primary keys, foreign keys, and appropriate data types for each attribute. - Normalize the database to eliminate redundancy and ensure data integrity.

3. Establish Database Connection: -

- Use appropriate programming language (such as PHP) to establish a connection between the server-side code and the database.
- Provide the necessary credentials (database host, username, password, database name) to establish the connection.

4. Execute SQL Queries: -

Use SQL (Structured Query Language) to write and execute queries for database operations such as data insertion, retrieval, updating, and deletion. - Sanitize user inputs to prevent SQL injection attacks by using prepared statements or parameterized queries.

5. Ticket Booking and Management: -

Implement database operations for ticket bookings, storing relevant details such as ticket type, departure date, arrival date, seat number, fare, and associated user details. - Retrieve and display booking details for users, manage cancellations or modifications by updating the corresponding database records.

6. Database Backup and Maintenance: -

Regularly backup the database to prevent data loss in case of server failures. - Implement scheduled tasks for database maintenance, including optimizing queries, indexing, and ensuring data consistency.

CODE SNIPPETS

1- To connect the database to website

```
1  <?php
2
3
4  try {
5      //host
6      define("HOST", "localhost");
7
8      //dbname
9      define("DBNAME", "wooxtravel");
10
11     //username
12     define("USER", "root");
13
14     //pass
15     define("PASS", "");
16
17
18     $conn = new PDO("mysql:host=".HOST.";dbname=".DBNAME."", USER, PASS);
19     $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
20
21
22 } catch( PDOException $Exception ) {
23     echo $Exception->getMessage();
24 }
25
```

2-Select Queries

```
$id = $_GET['id'];

$country = $conn->query("SELECT * FROM countries WHERE id='$id'");
$country->execute();

$singleCountry = $country->fetch(PDO::FETCH_OBJ);

//images for the cities
$citiesImages = $conn->query("SELECT * FROM cities WHERE country_id='$id'");
$citiesImages->execute();

$singleImage = $citiesImages->fetchAll(PDO::FETCH_OBJ);
```

3-Join Queries

```
//cities with bookings number

$cities = $conn->query("SELECT cities.id AS id, cities.name
AS name, cities.image AS image, cities.trip_days AS trip_days, cities.price AS price,
COUNT(bookings.city_id) AS count_bookings FROM cities LEFT JOIN
bookings ON cities.id = bookings.city_id WHERE cities.country_id = '$id' GROUP BY(bookings.city_id)");

$cities->execute();

$sallCities = $cities->fetchAll(PDO::FETCH_OBJ);

//cities of every country

$cities_country = $conn->query("SELECT COUNT(country_id) AS num_city FROM cities
WHERE country_id = '$id'");
$cities_country->execute();

$num_cities = $cities_country->fetch(PDO::FETCH_OBJ);

//number of bookings for every country

$num_country = $conn->query("SELECT COUNT(bookings.city_id) AS count_bookings
FROM cities JOIN bookings
ON cities.id = bookings.city_id WHERE cities.country_id = '$id'");
$num_country->execute();

$num_bookings = $num_country->fetch(PDO::FETCH_OBJ);
```

4-Code to search/filter the Deals

```
$country_id = $_POST['country_id'];
$price = $_POST['price'];

$searchs = $conn->query("SELECT * FROM cities WHERE country_id = $country_id
AND price < $price");
$searchs->execute();

$searchs = $searchs->fetchAll(PDO::FETCH_OBJ);
```


5-Insertion of Bookings

```
$name = $_POST['name'];
$phone_number = $_POST['phone_number'];
$num_of_geusts = $_POST['num_of_geusts'];
$checkin_date = $_POST['checkin_date'];
$destination = $_POST['destination'];
$status = "Pending";
$city_id = $id;
$user_id = $_SESSION['user_id'];

$payment = $num_of_geusts * $getCity->price;

$_SESSION['payment'] = $payment;

if(date("Y-m-d") < $checkin_date) {
    $insert = $conn->prepare("INSERT INTO bookings (name, phone_number, num_of_geusts,
    checkin_date, destination, status, city_id, user_id, payment)
    VALUES (:name, :phone_number, :num_of_geusts, :checkin_date, :destination, :status,
    :city_id, :user_id, :payment)");

    $insert->execute([
        ":name" => $name,
        ":phone_number" => $phone_number,
        ":num_of_geusts" => $num_of_geusts,
        ":checkin_date" => $checkin_date,
        ":destination" => $destination,
        ":status" => $status,
        ":city_id" => $city_id,
        ":user_id" => $user_id,
        ":payment" => $payment,
    ]);
}
```

6-Booking Status Updation

```
if(isset($_GET['id']) AND isset($_GET['status'])) {

    $id = $_GET['id'];
    $status = $_GET['status'];

    if($status == "Pending") {
        $update = $conn->prepare("UPDATE bookings SET status='Booked Successfully' WHERE
        id='$id'");

        $update->execute();

        header("location: show-bookings.php");
    } else {
        $update = $conn->prepare("UPDATE bookings SET status='Pending' WHERE
        id='$id'");

        $update->execute();
    }
}
```


7-To delete city records(by Admin)

```
if(!isset($_SESSION["adminname"])) {  
    header("location: ".ADMINURL."");  
}  
  
if(isset($_GET['id'])) {  
  
    $id = $_GET['id'];  
  
    $image_delete = $conn->query("SELECT * FROM cities WHERE id='$id'");  
    $image_delete->execute();  
  
    $getImage = $image_delete->fetch(PDO::FETCH_OBJ);  
  
    unlink("images_cities/" . $getImage->image);  
  
    //deleteCitiesRecord  
  
    $deleteRecord = $conn->query("DELETE FROM cities WHERE id='$id'");  
    $deleteRecord->execute();  
  
    header("location: show-cities.php");  
}
```

Conclusion

A Tour Management System (TMS) is a web-based application that automates travel-related processes. It can be used by travel agencies and companies to manage their operations.

A TMS can help with:

- Booking trips
- Managing upcoming trips
- Tracking and reporting on previous trips
- Automating corporate travel policies
- Consolidating travel invoices and vendors
- Handling all aspects of travel, such as scheduling and excursions
- Linking consumers and agents directly
- Providing a feedback method for tourists
- Managing and regulating the collection of tourists' information

By leveraging HTML, CSS, PHP, and MySQL, the project enables seamless integration of the front-end interface, server-side processing, and database management. This ensures the reliability, scalability, and security of the system. Overall, the tour management system significantly improves the efficiency and convenience of tour bookings, enhances user experience, and simplifies the administrative tasks for travel agencies. It promotes a more organized and streamlined approach to travel management, benefiting both travelers and service providers alike.

Appendix

- Tools used:

For FrontEnd - HTML & CSS (BootStrap & jQuery)

For Backend - PHP, MySQL & PDO

For connecting database with the website- phpMyAdmin

For connecting both - PHP (PDO)

Platform - VS Code

For Entity Relationship Diagram - Draw.io

- References

https://www.w3schools.com/html/html_css.asp

<https://developer.mozilla.org/en-US/>

<https://www.w3schools.com/php/>