Mini project report on

Online Travel Portal System

Submitted in partial fulfilment of the requirements for the award of degree of

Bachelor of Technology

in

Computer Science & Engineering

UE20CS301 – DBMS Project

Submitted by:

Neeraja N   **PES2UG20CS528**

Under the guidance of

Prof. Nivedita Kasturi

Assistant Professor

Designation

PES University

AUG - DEC 2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

FACULTY OF ENGINEERING

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

CERTIFICATE

This is to certify that the mini project entitled

Online Travel Portal System

is a bonafide work carried out by

Neeraja N   **PES2UG20CS528**

In partial fulfilment for the completion of fifth semester DBMS Project (UE20CSS301) in the Program of Study - Bachelor of Technology in Computer Science and Engineering under rules and regulations of PES University, Bengaluru during the period AUG. 2022 – DEC. 2022. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report. The project has been approved as it satisfies the 5th semester academic requirements in respect of project work.

Signature

Prof. Nivedita Kasturi

Assistant Professor

DECLARATION

We hereby declare that the DBMS Project entitled **Online Travel Portal System** has been carried out by us under the guidance of **Prof. Nivedita Kasturi, Assistant Professor** and submitted in partial fulfilment of the course requirements for the award of degree of **Bachelor of Technology** in **Computer Science and Engineering** of **PES University, Bengaluru** during the academic semester AUG – DEC 2022.

Neeraja N   **PES2UG20CS528**   **<Signature>**

ACKNOWLEDGEMENT

ABSTRACT

Online Travel Portal System provides a platform from where any customer, agent, or enterpriser can do Flight Booking, Hotel Booking, Car Booking, Bus Booking, or Holiday Package Booking.

Today, travel has become a part of life because it plays a vital role in our daily needs. In this modern age, people want to do everything with just a few mouse clicks and hence this travel portal provides a comfortable and User-friendly platform for online travel agencies and customers.

TABLE OF CONTENTS

1. INTRODUCTION

Travel Portal System provides the customers with the best experience at their finger touch using an advanced travel app to make their travel experience convenient and memorable. A pleasant travel plan includes booking from mobile travel apps such as Flight Tickets, Car Vouchers, Bus Vouchers, Hotel Vouchers, or Holiday Booking, and many more.

Travel Portal Development has attracted many small businesses and offered them the opportunity to earn great profits. With Travel Portal development, you get a booking engine system for Flights, Hotels, cars, and buses. Customer Management System, Bookings and Cancellation of Flight Tickets, Bus Voucher, Car and Bus Vouchers, various offers under Customer Logins, Payment Gateway Integration, Account Management System and many more features. It is very certain that Travel Portal Development is an important tool to enhance your business. It is considered one of the most important aspects of the travel business.

2. PROBLEM DEFINITION

Travel portal System automate our process of the travel business and provide good distribution. Customers can compare the prices of holiday/trip packages offered by many travel agencies having their own T_ids (travel ids). Customers can choose the type of the trip, day of trip, book any seat type of their wish, reserve a room in hotel from wide range of options available from many travel agencies. When a customer login and make a booking, his account is assigned to an employee. Each employee from a specific travel agency manages the account of a set of customers under him by providing him options to update, cancel or create a booking.

3. ER MODEL
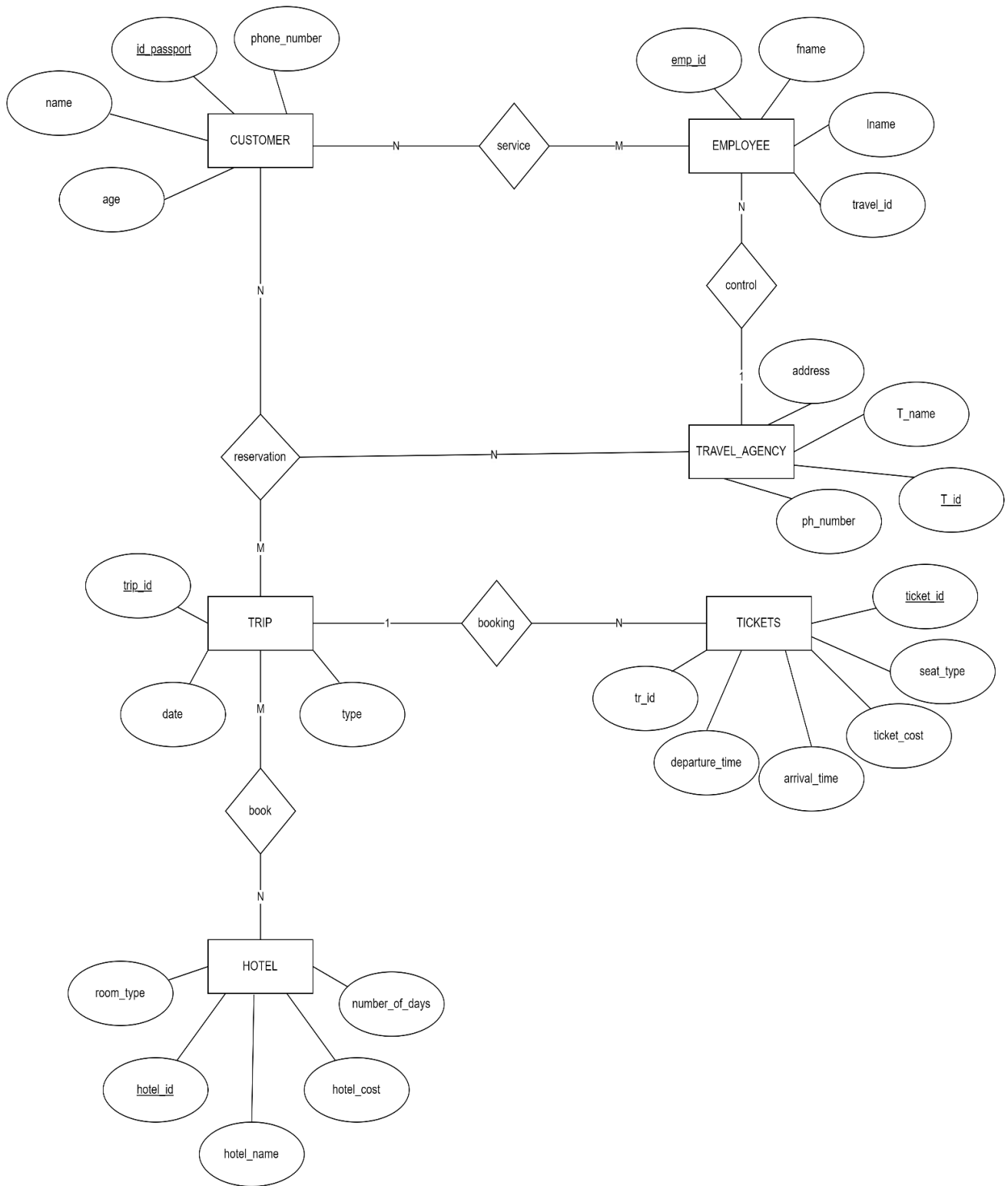
4. ER TO RELATIONAL MAPPING

4.1 STEPS OF ALGORITHM FOR CHOOSEN PROBLEM

Step 1: Mapping of Regular Entity Types

CUSTOMER

id_passport	name	age	phone_number

EMPLOYEE

emp_id	fname	lname	travel_id

TRAVEL_AGENCY

T_id	T_name	address	ph_number

TRIP

trip_id	type	date

TICKETS

ticket_id	seat_type	ticket_cost	arrival_time	departure_time	tr_id

HOTEL

hotel_id	room_type	hotel_name	hotel_cost	number_of_days

Step 2: Mapping of Weak Entity Types

There are no weak entity types

Step 3: Mapping of Binary 1:1 Relation Types

There are no 1:1 relation types

Step 4: Mapping of Binary 1:N Relationship Types.

TRAVEL_AGENCY

T_id	T_name	address	ph_number

EMPLOYEE

emp_id	fname	lname	travel_id

TRIP

trip_id	type	date

TICKETS

ticket_id	seat_type	ticket_cost	arrival_time	departure_time	tr_id

Step 5: Mapping of Binary M:N Relationship Types.

CUSTOMER

id_passport	name	age	phone_number

TRAVEL_AGENCY

T_id	T_name	address	ph_number

EMPLOYEE

emp_id	fname	lname	travel_id

TRIP

trip_id	type	date

TICKETS

ticket_id	seat_type	ticket_cost	arrival_time	departure_time	tr_id

HOTEL

hotel_cost	room_type	hotel_name	hotel_id	number_of_days

SERVICE

pass_id	employee_id

BOOK

id_hotel	tripid
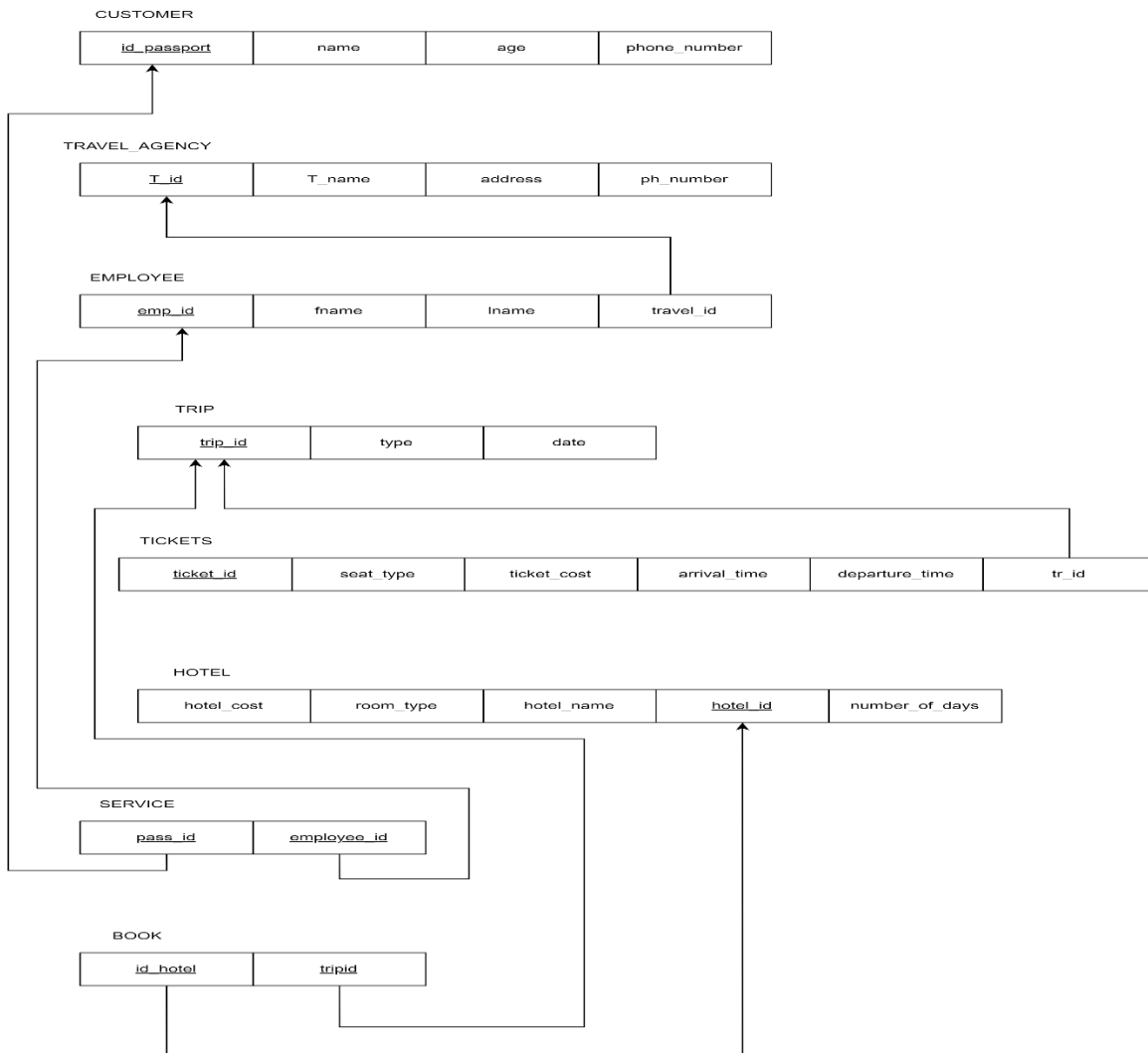
Step 6: Mapping of Multivalued attributes.

There are no multivalued attributes.

Step 7: Mapping of N-ary Relationship Types.

CUSTOMER

id_passport	name	age	phone_number

TRAVEL_AGENCY

ph_number	T_name	address	T_id

TRIP

date	type	trip_id

RESERVATION

passport_id	travel_id	tripid
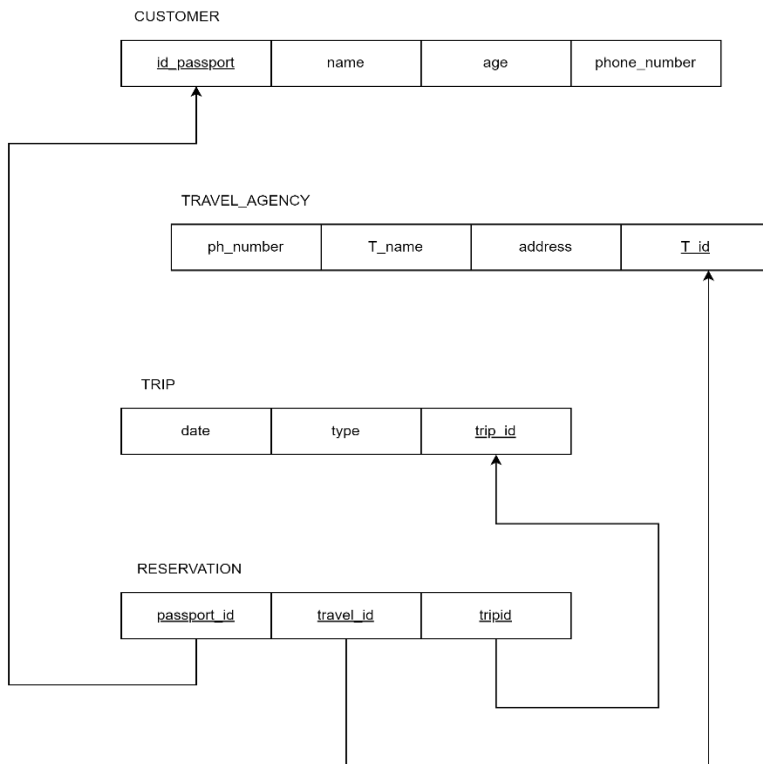
4.2 COMPLETE DIAGRAM OF RELATIONAL MAPPING

CUSTOMER

PK	id_passport
	phone_number
	name
	age

reservation

passport_id
travel_id
tripid

TICKETS

PK	ticket_id
	seat_type
	ticket_cost
	arrival_time
	departure_time
	tr_id

service

pass_id
employee_id

TRIP

PK	trip_id
	date
	type

HOTEL

PK	hotel_id
	number_of_days
	hotel_cost
	hotel_name
	room_type

EMPLOYEE

PK	emp_id
	fname
	lname
	travel_id

TRAVEL_AGENCY

PK	T_id
	ph_number
	address
	T_name

book

tripid
id_hotel

5. DDL STATEMENTS

STATEMENTS WITH SCREEN SHOTS OF THE TABLE CREATION

1. Table customer

CREATE TABLE `customer` (

  `phone_number` int(11) NOT NULL,

  `id_passport` int(5) NOT NULL,

  `name` varchar(20) NOT NULL,

  `age` int(11) NOT NULL

)

ALTER TABLE `customer`

  ADD PRIMARY KEY (`id_passport`);

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
1	phone_number	int(11)			No	*None*			🖉 Change	🚫 Drop	More
2	id_passport 🔑	int(5)			No	*None*			🖉 Change	🚫 Drop	More
3	name	varchar(20)	utf8mb4_general_ci		No	*None*			🖉 Change	🚫 Drop	More
4	age	int(11)			No	*None*			🖉 Change	🚫 Drop	More

2. Table employee

CREATE TABLE `employee` (

  `emp_id` int(8) NOT NULL,

  `fname` varchar(15) NOT NULL,

  `lname` varchar(15) NOT NULL,

  `travel_id` int(5) NOT NULL

)

ALTER TABLE `employee`

  ADD PRIMARY KEY (`emp_id`),

  ADD KEY `fk_emp_travel` (`travel_id`);

ALTER TABLE `employee`
ADD CONSTRAINT `fk_emp_travel` FOREIGN KEY (`travel_id`) REFERENCES `travel_agency` (`T_id`);

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
1	emp_id 🔑	int(8)			No	None			🖉 Change	⊖ Drop	More
2	fname	varchar(15)	utf8mb4_general_ci		No	None			🖉 Change	⊖ Drop	More
3	lname	varchar(15)	utf8mb4_general_ci		No	None			🖉 Change	⊖ Drop	More
4	travel_id 🔑	int(5)			No	None			🖉 Change	⊖ Drop	More

3. Table hotel

CREATE TABLE `hotel` (
  `room_type` char(1) NOT NULL,
  `hotel_id` int(3) NOT NULL,
  `hotel_name` varchar(20) NOT NULL,
  `hotel_cost` decimal(10,0) NOT NULL,
  `number_of_days` int(11) NOT NULL
)
ALTER TABLE `hotel`
ADD PRIMARY KEY (`hotel_id`);

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
1	room_type	char(1)	utf8mb4_general_ci		No	None			🖉 Change	⊖ Drop	More
2	hotel_id 🔑	int(3)			No	None			🖉 Change	⊖ Drop	More
3	hotel_name	varchar(20)	utf8mb4_general_ci		No	None			🖉 Change	⊖ Drop	More
4	hotel_cost	decimal(10,0)			No	None			🖉 Change	⊖ Drop	More
5	number_of_days	int(11)			No	None			🖉 Change	⊖ Drop	More

4. Table tickets

CREATE TABLE `tickets` (

```
  `ticket_id` int(5) NOT NULL,

  `seat_type` char(1) NOT NULL,

  `ticket_cost` decimal(10,0) NOT NULL,

  `arrival_time` time NOT NULL,

  `departure_time` time NOT NULL,

  `tr_id` int(5) NOT NULL

)
ALTER TABLE `tickets`
  ADD PRIMARY KEY (`ticket_id`),
  ADD KEY `fk_trip_tickets` (`tr_id`);
ALTER TABLE `tickets`
  ADD CONSTRAINT `fk_trip_tickets` FOREIGN KEY (`tr_id`) REFERENCES `trip`
(`trip_id`);
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
1	ticket_id 🔑	int(5)			No	None			🖉 Change	⊖ Drop	More
2	seat_type	char(1)	utf8mb4_general_ci		No	None			🖉 Change	⊖ Drop	More
3	ticket_cost	decimal(10,0)			No	None			🖉 Change	⊖ Drop	More
4	arrival_time	time			No	None			🖉 Change	⊖ Drop	More
5	departure_time	time			No	None			🖉 Change	⊖ Drop	More
6	tr_id 🔑	int(5)			No	None			🖉 Change	⊖ Drop	More

5. Table travel_agency

```
CREATE TABLE `travel_agency` (
  `address` text NOT NULL,
  `ph_number` int(11) NOT NULL,
  `T_name` varchar(15) NOT NULL,
  `T_id` int(5) NOT NULL
)
ALTER TABLE `travel_agency`
  ADD PRIMARY KEY (`T_id`);
```

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
1	**address**	text	utf8mb4_general_ci		No	*None*			🖊 Change	⛔ Drop	More
2	**ph_number**	int(11)			No	*None*			🖊 Change	⛔ Drop	More
3	**T_name**	varchar(15)	utf8mb4_general_ci		No	*None*			🖊 Change	⛔ Drop	More
4	**T_id** 🔑	int(5)			No	*None*			🖊 Change	⛔ Drop	More

6. Table trip

CREATE TABLE `trip` (

  `trip_id` int(5) NOT NULL,

  `date` date NOT NULL,

  `type` varchar(20) NOT NULL

)

ALTER TABLE `trip`

  ADD PRIMARY KEY (`trip_id`);

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
1	**trip_id** 🔑	int(5)			No	*None*			🖊 Change	⛔ Drop	More
2	**date**	date			No	*None*			🖊 Change	⛔ Drop	More
3	**type**	varchar(20)	utf8mb4_general_ci		No	*None*			🖊 Change	⛔ Drop	More

7. Table book

CREATE TABLE `book` (

  `tripid` int(5) NOT NULL,

  `id_hotel` int(3) NOT NULL

)

ALTER TABLE `book`

  ADD CONSTRAINT `fk_book_hotel` FOREIGN KEY (`id_hotel`) REFERENCES `hotel`

(`hotel_id`),

ADD CONSTRAINT `fk_book_trip` FOREIGN KEY (`tripid`) REFERENCES `trip` (`trip_id`);

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
1	tripid 🔑	int(5)			No	None			🖉 Change	⊖ Drop	More
2	id_hotel 🔑	int(3)			No	None			🖉 Change	⊖ Drop	More

8. Table reservation

CREATE TABLE `reservation` (
  `passport_id` int(5) NOT NULL,
  `travel_id` int(5) NOT NULL,
  `tripid` int(5) NOT NULL
)
ALTER TABLE `reservation`
  ADD CONSTRAINT `fk_cust_reserve` FOREIGN KEY (`passport_id`) REFERENCES `customer` (`id_passport`),
  ADD CONSTRAINT `fk_travel_reserve` FOREIGN KEY (`travel_id`) REFERENCES `travel_agency` (`T_id`),
  ADD CONSTRAINT `fk_trip_reserve` FOREIGN KEY (`tripid`) REFERENCES `trip` (`trip_id`);

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
1	passport_id 🔑	int(5)			No	None			🖉 Change	⊖ Drop	More
2	travel_id 🔑	int(5)			No	None			🖉 Change	⊖ Drop	More
3	tripid 🔑	int(5)			No	None			🖉 Change	⊖ Drop	More

9. Table service

CREATE TABLE `service` (
  `pass_id` int(5) NOT NULL,
  `employee_id` int(8) NOT NULL

)

ALTER TABLE `service`

  ADD CONSTRAINT `fk_cust_service` FOREIGN KEY (`pass_id`) REFERENCES `customer` (`id_passport`),

  ADD CONSTRAINT `fk_employee_service` FOREIGN KEY (`employee_id`) REFERENCES `employee` (`emp_id`);

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
1	pass_id 🔑	int(5)			No	*None*			🖉 Change	⊖ Drop	More
2	employee_id 🔑	int(8)			No	*None*			🖉 Change	⊖ Drop	More

6. DML STATEMENTS

STATEMENTS WITH SCREEN SHOTS OF THE TABLE WITH INSERTED VALUES

1. Table customer

INSERT INTO `customer` (`phone_number`, `id_passport`, `name`, `age`) VALUES

(2125987413, 12356, 'Noah', 35),

(2135457412, 56987, 'James', 12),

(2125698745, 58419, 'Joseph', 25),

(1123657845, 65478, 'Benjamin', 36),

(1256897423, 89654, 'Alexander', 20);

←T→				phone_number	id_passport	name	age
☐	🖊 Edit	🔢 Copy	⊝ Delete	2125987413	12356	Noah	35
☐	🖊 Edit	🔢 Copy	⊝ Delete	2135457412	56987	James	19
☐	🖊 Edit	🔢 Copy	⊝ Delete	2125698745	58419	Joseph	25
☐	🖊 Edit	🔢 Copy	⊝ Delete	1123657845	65478	Benjamin	36
☐	🖊 Edit	🔢 Copy	⊝ Delete	1256897423	89654	Alexander	20

2. Table employee

INSERT INTO `employee` (`emp_id`, `fname`, `lname`, `travel_id`) VALUES

(45698736, 'Elizabeth', 'A', 11232),

(56321789, 'George', 'C', 34762),

(58741236, 'Micheal', 'B', 12346),

(65874139, 'Anna', 'M', 59874),

(65897412, 'Grace', 'A', 68714);

←T→				emp_id	fname	lname	travel_id
☐	✏ Edit	⧉ Copy	⊖ Delete	45698736	Elizabeth	A	11232
☐	✏ Edit	⧉ Copy	⊖ Delete	56321789	George	C	34762
☐	✏ Edit	⧉ Copy	⊖ Delete	58741236	Micheal	B	12346
☐	✏ Edit	⧉ Copy	⊖ Delete	65874139	Anna	M	59874
☐	✏ Edit	⧉ Copy	⊖ Delete	65897412	Grace	A	68714

3. Table hotel

INSERT INTO `hotel` (`room_type`, `hotel_id`, `hotel_name`, `hotel_cost`, `number_of_days`) VALUES

('A', 123, 'Holiday Inn', '5500', 2),

('B', 179, 'Grand hotel', '14000', 4),

('A', 235, 'Crown Plaza', '15000', 6),

('B', 569, 'Hotel Indigo', '7526', 5),

('C', 784, 'Four seasons', '10000', 5);

←T→				room_type	hotel_id	hotel_name	hotel_cost	number_of_days
☐	✏ Edit	⧉ Copy	⊖ Delete	A	123	Holiday Inn	5500	2
☐	✏ Edit	⧉ Copy	⊖ Delete	B	179	Grand hotel	14000	4
☐	✏ Edit	⧉ Copy	⊖ Delete	A	235	Crown Plaza	15000	6
☐	✏ Edit	⧉ Copy	⊖ Delete	B	569	Hotel Indigo	7526	5
☐	✏ Edit	⧉ Copy	⊖ Delete	C	784	Four seasons	10000	5

4. Table tickets

INSERT INTO `tickets` (`ticket_id`, `seat_type`, `ticket_cost`, `arrival_time`, `departure_time`, `tr_id`) VALUES

(12345, 'B', '5000', '09:00:00', '09:20:00', 11233),

(45781, 'A', '25600', '10:30:00', '10:45:00', 12121),

(58471, 'A', '2500', '08:00:00', '08:10:00', 45789),

(58743, 'A', '1500', '11:00:00', '11:30:00', 54236),

(63547, 'B', '4500', '11:15:00', '11:20:00', 54712);

	ticket_id	seat_type	ticket_cost	arrival_time	departure_time	tr_id
☐ 🖉 Edit 🛃 Copy ⊖ Delete	12345	B	5000	09:00:00	09:20:00	11233
☐ 🖉 Edit 🛃 Copy ⊖ Delete	45781	A	25600	10:30:00	10:45:00	12121
☐ 🖉 Edit 🛃 Copy ⊖ Delete	58471	A	2500	08:00:00	08:10:00	45789
☐ 🖉 Edit 🛃 Copy ⊖ Delete	58743	A	1500	11:00:00	11:30:00	54236
☐ 🖉 Edit 🛃 Copy ⊖ Delete	63547	B	4500	11:15:00	11:20:00	54712

5. Table travel_agency

INSERT INTO `travel_agency` (`address`, `ph_number`, `T_name`, `T_id`) VALUES
('No:12,BTM layout,Bangalore', 2136589467, 'Oasis Travel', 11232),
('No:16,Shree layout,Kerala', 1269874563, 'Paradise travel', 12346),
('No:21,Chandra layout,Mangalore', 2137634112, 'Sunset vacation', 34762),
('No:56,Vikas layout,Delhi', 2119874365, 'Vacation connec', 59874),
('No:24,Reddy layout,Bangalore', 1987456379, 'Tropical travel', 68714);

	address	ph_number	T_name	T_id
☐ 🖉 Edit 🛃 Copy ⊖ Delete	No:12,BTM layout,Bangalore	2136589467	Oasis Travel	11232
☐ 🖉 Edit 🛃 Copy ⊖ Delete	No:16,Shree layout,Kerala	1269874563	Paradise travel	12346
☐ 🖉 Edit 🛃 Copy ⊖ Delete	No:21,Chandra layout,Mangalore	2137634112	Sunset vacation	34762
☐ 🖉 Edit 🛃 Copy ⊖ Delete	No:56,Vikas layout,Delhi	2119874365	Vacation connec	59874
☐ 🖉 Edit 🛃 Copy ⊖ Delete	No:24,Reddy layout,Bangalore	1987456379	Tropical travel	68714

6. Table trip

INSERT INTO `trip` (`trip_id`, `date`, `type`) VALUES
(11233, '2022-12-15', 'Beach'),
(12121, '2023-02-08', 'Eco tour'),
(45789, '2023-01-10', 'Adventure'),
(54236, '2022-11-03', 'Medical'),
(54712, '2022-12-30', 'Adventure');

←T→				trip_id	date	type
☐	🖉 Edit	📋 Copy	⊖ Delete	11233	2022-12-15	Beach
☐	🖉 Edit	📋 Copy	⊖ Delete	12121	2023-02-08	Eco tour
☐	🖉 Edit	📋 Copy	⊖ Delete	45789	2023-01-10	Adventure
☐	🖉 Edit	📋 Copy	⊖ Delete	54236	2022-11-03	Medical
☐	🖉 Edit	📋 Copy	⊖ Delete	54712	2022-12-30	Adventure

7. Table book

INSERT INTO `book` (`tripid`, `id_hotel`) VALUES

(11233, 179),

(54712, 123),

(11233, 179),

(54236, 569),

(54712, 569);

tripid	id_hotel
11233	179
54712	123
11233	179
54236	569
54712	569

8. Table reservation

INSERT INTO `reservation` (`passport_id`, `travel_id`, `tripid`) VALUES

(65478, 12346, 54712),

(89654, 11232, 11233),

(56987, 34762, 45789),

(58419, 59874, 12121),

(12356, 68714, 45789);

passport_id	travel_id	tripid
65478	12346	54712
89654	11232	11233
56987	34762	45789
58419	59874	12121
12356	68714	45789

9. **Table service**

INSERT INTO `service` (`pass_id`, `employee_id`) VALUES

(12356, 45698736),

(56987, 56321789),

(58419, 58741236),

(65478, 65874139),

(89654, 65897412);

pass_id	employee_id
12356	45698736
56987	56321789
58419	58741236
65478	65874139
89654	65897412

7. QUERIES

a. One Nested Query and one Correlated query

Retrieve the names of customers who is going on adventure trip

Select name from customer where id_passport IN
(Select passport_id from reservation where tripid IN
(SELECT trip_id from trip where type="adventure"));

✔ Showing rows 0 - 2 (3 total, Query took 0.0007 seconds.)

Select name from customer where id_passport IN (Select passport_id from reservation where tripid IN (SELECT trip_id from trip where type="adventure"));

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 ∨ Filter rows: Search this table Sort by key: None ∨

Extra options

←T→				name
☐	✎ Edit	⌗ Copy	⊖ Delete	Noah
☐	✎ Edit	⌗ Copy	⊖ Delete	James
☐	✎ Edit	⌗ Copy	⊖ Delete	Benjamin

Retrieve the names of customers having tripid=12121

Select name from customer C where EXISTS
(select * from reservation R where C.id_passport=R.passport_id and R.tripid='12121');

✔ Showing rows 0 - 0 (1 total, Query took 0.0008 seconds.)

Select name from customer C where EXISTS (select * from reservation R where C.id_passport=R.passport_id and R.tripid='12121');

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 ∨ Filter rows: Search this table

Extra options

←T→				name
☐	✎ Edit	⌗ Copy	⊖ Delete	Joseph

b. Two queries for SET Operations

Retrieve the trip_id of trip scheduled on either of the dates 10/01/2023 or 30/12/2022 or both with adventure type

Select trip_id from trip where (date="2023-01-10" and type="Adventure")
UNION
select trip_id from trip where (date="2022-12-30" and type="Adventure");

trip_id
45789
54712

Retrieve the travel ids present in both tables tickets and travel_agency
select tr_id from tickets
where EXISTS
(select T_id from travel_agency);

←T→				tr_id
☐	✏ Edit	⊒ᵢ Copy	⊖ Delete	11233
☐	✏ Edit	⊒ᵢ Copy	⊖ Delete	12121
☐	✏ Edit	⊒ᵢ Copy	⊖ Delete	45789
☐	✏ Edit	⊒ᵢ Copy	⊖ Delete	54236
☐	✏ Edit	⊒ᵢ Copy	⊖ Delete	54712

c. Two queries for Aggregate functions with group by clause, use HavingClause.

Get the number of tickets booked in each seat type
SELECT seat_type, COUNT(ticket_id) from tickets
group by seat_type
having COUNT(ticket_id);

seat_type	COUNT(ticket_id)
A	3
B	2

Retrieve the minimum cost for each room type
select room_type, min(hotel_cost) from hotel
group by room_type
having min(hotel_cost);

			room_type	min(hotel_cost)
☐	🖉 Edit	⌗ Copy ⊖ Delete	A	5500
☐	🖉 Edit	⌗ Copy ⊖ Delete	B	7526
☐	🖉 Edit	⌗ Copy ⊖ Delete	C	10000

d. Two Simple Queries with order by clause.

Sort the customer table in descending order by name

SELECT * from customer

order by name desc;

			phone_number	id_passport	name ▼ 1	age
☐	🖉 Edit	⌗ Copy ⊖ Delete	2125987413	12356	Noah	35
☐	🖉 Edit	⌗ Copy ⊖ Delete	2125698745	58419	Joseph	25
☐	🖉 Edit	⌗ Copy ⊖ Delete	2135457412	56987	James	20
☐	🖉 Edit	⌗ Copy ⊖ Delete	1123657845	65478	Benjamin	36
☐	🖉 Edit	⌗ Copy ⊖ Delete	1256897423	89654	Alexander	20

Sort the hotels in ascending order of their costs

SELECT * from hotel

order by hotel_cost ASC;

			room_type	hotel_id	hotel_name	hotel_cost ▲ 1	number_of_days
☐	🖉 Edit	⌗ Copy ⊖ Delete	A	123	Holiday Inn	5500	2
☐	🖉 Edit	⌗ Copy ⊖ Delete	B	569	Hotel Indigo	7526	5
☐	🖉 Edit	⌗ Copy ⊖ Delete	C	784	Four seasons	10000	5
☐	🖉 Edit	⌗ Copy ⊖ Delete	B	179	Grand hotel	14000	4
☐	🖉 Edit	⌗ Copy ⊖ Delete	A	235	Crown Plaza	15000	6

8. STORED PROCEDURES, FUCNTIONS AND TRIGGERS

8.1 STORED PROCEDURES OR FUNCTIONS

Retrieve the details of the customers whose age is above 25 by creating a procedure.

```
DELIMITER &&
CREATE PROCEDURE get_age_above_25 ()
BEGIN
    SELECT * FROM customer WHERE age > 25;
END &&
DELIMITER ;
```

```
CALL get_age_above_25();
```

✔ Showing rows 0 - 1 (2 total, Query took 0.0004 seconds.)

```
CALL get_age_above_25();
```

[Edit inline] [Edit] [Create PHP code]

☐ Show all | Number of rows: 25 ⌄   Filter rows: Search this table

Extra options

phone_number	id_passport	name	age
2125987413	12356	Noah	35
1123657845	65478	Benjamin	36

8.2 TRIGGERS

Create a trigger to stop updating the customer table if new value of the age entered is less than 18 years.

```sql
DELIMITER $$
CREATE TRIGGER before_update_customer
BEFORE UPDATE
ON customer FOR EACH ROW
BEGIN
    DECLARE error_msg VARCHAR(255);
    SET error_msg = ('Age of the customer cannot be less than 18 years');
        IF new.age < 18 THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = error_msg;
    END IF;
END $$
DELIMITER ;


UPDATE customer
set age=15
where id_passport=56987;
```

Error

SQL query: Copy

```
UPDATE customer
set age=15
where id_passport=56987;
```

MySQL said: 🔵

`#1644 - Age of the customer cannot be less than 18 years`

UPDATE customer

set age=20

where id_passport=56987;

✔ 1 row affected. (Query took 0.0037 seconds.)

UPDATE customer set age=20 where id_passport=56987;

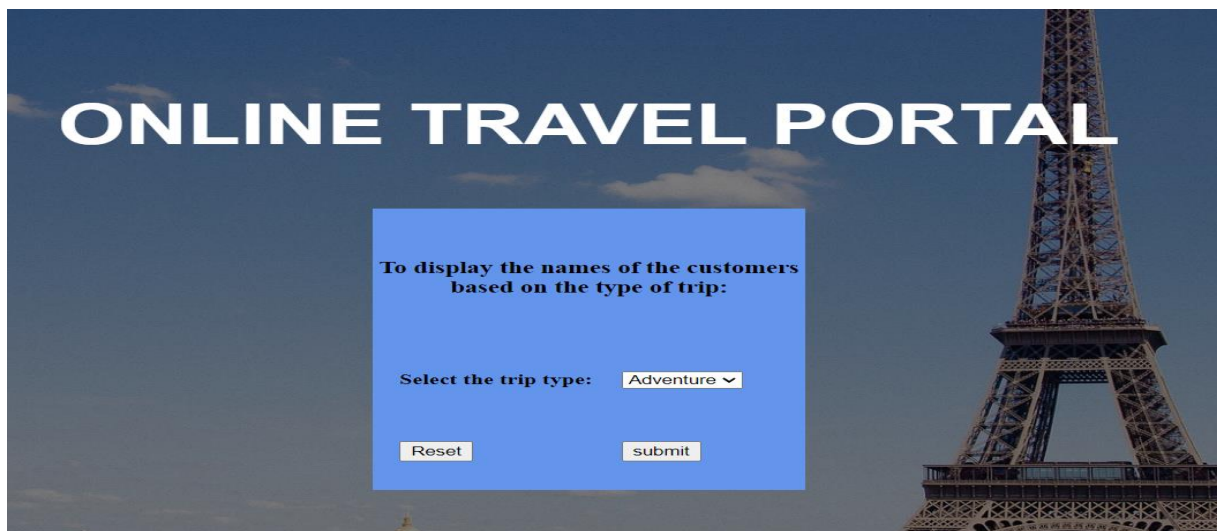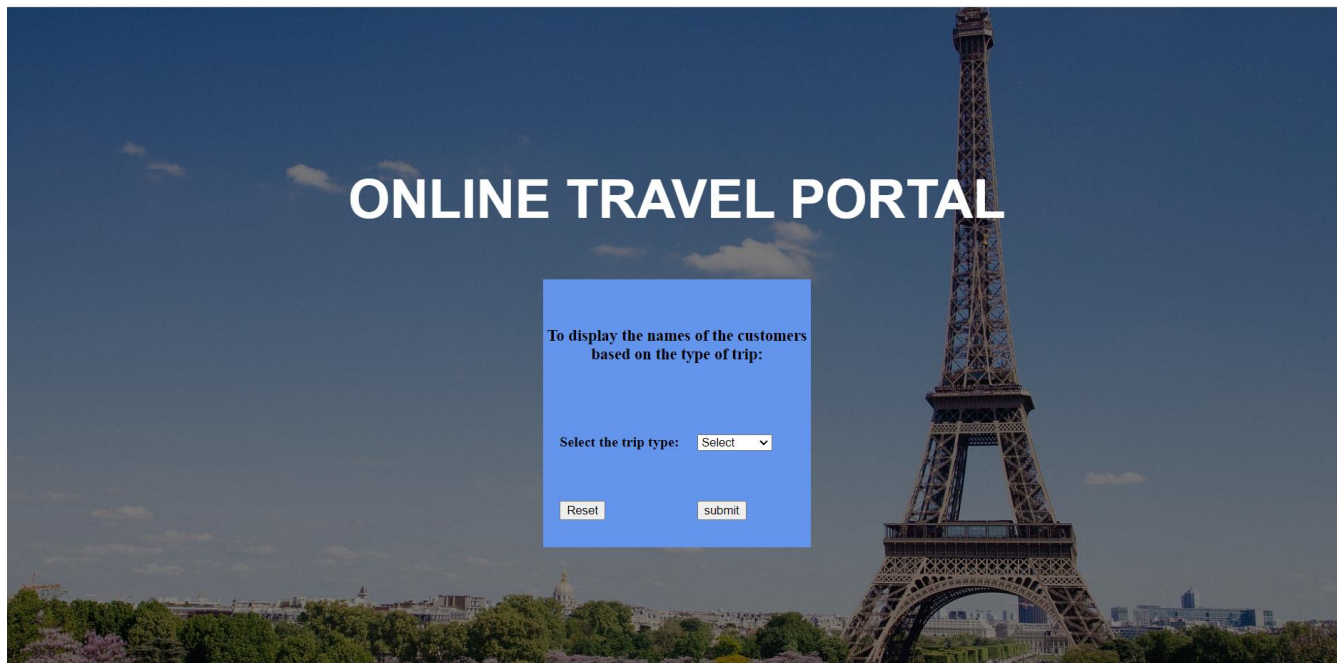[Edit inline] [Edit] [Create PHP code]

9. FRONT END DEVELOPEMNT

Developing frontend for the query:

Retrieve the names of customers who is going on adventure trip:

Select name from customer where id_passport IN

(Select passport_id from reservation where tripid IN

(SELECT trip_id from trip where type="adventure"));

Online travel portal system

Name and passport id of the cutomers who is going on 'Adventure' trip

Customer name	Passport id
Noah	12356
James	56987
Benjamin	65478

REFERENCES

1. https://www.freeprojectz.com/entity-relationship/online-travel-portal-system-er-diagram

2. https://www.geeksforgeeks.org/how-to-insert-form-data-into-database-using-php/