



PES UNIVERSITY, BANGALORE  
Department of Computer Science and Engineering

**B.TECH. (CSE)**  
**VI SEMESTER**  
**UE20CS301 – Database Management Systems (Minors)**  
**Mini-Project Report**  
**on**  
**Title**

SUBMITTED BY

Name	SRN
Alleti Rohan Reddy	PES1UG20ME009

**January – May 2023**  
**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**  
**BENGALURU – 560100, KARNATAKA, INDIA**



PES UNIVERSITY, BANGALORE  
Department of Computer Science and Engineering

## Project Synopsis

Title: Travel Management System

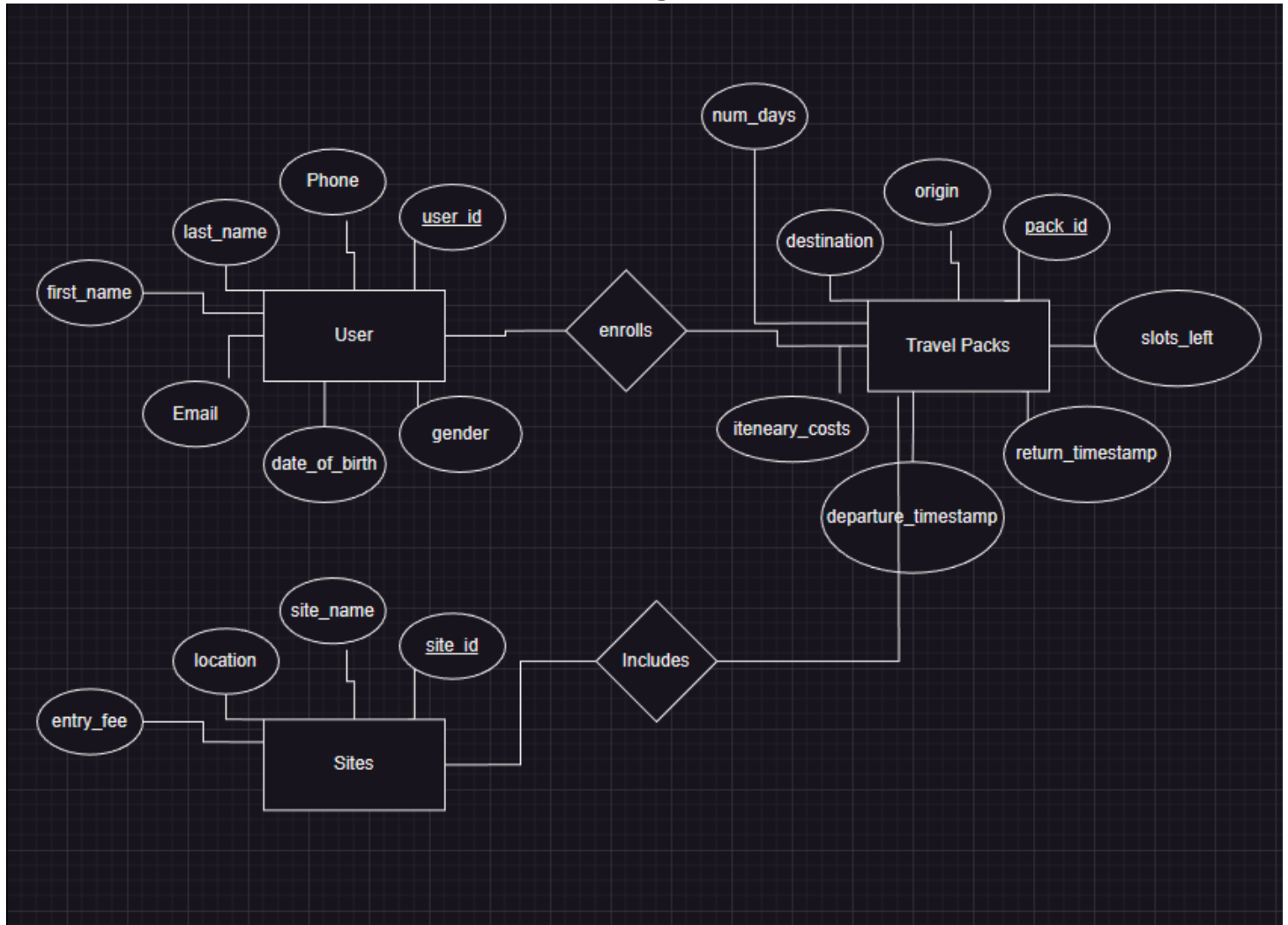
Introduction: A system designed to book vacations and travel destinations with ease through a web application.

It will cover the following functionalities:

- Collect and store user information.
- Create travel packs which users can choose from.
- Store user's choices of their travel packs.
- Each travel pack has limited slots.
- Customization of travel packs by adding or removing site locations in the destination.

Methodology: The application has been built using streamlit python for the frontend and mysql as the database.

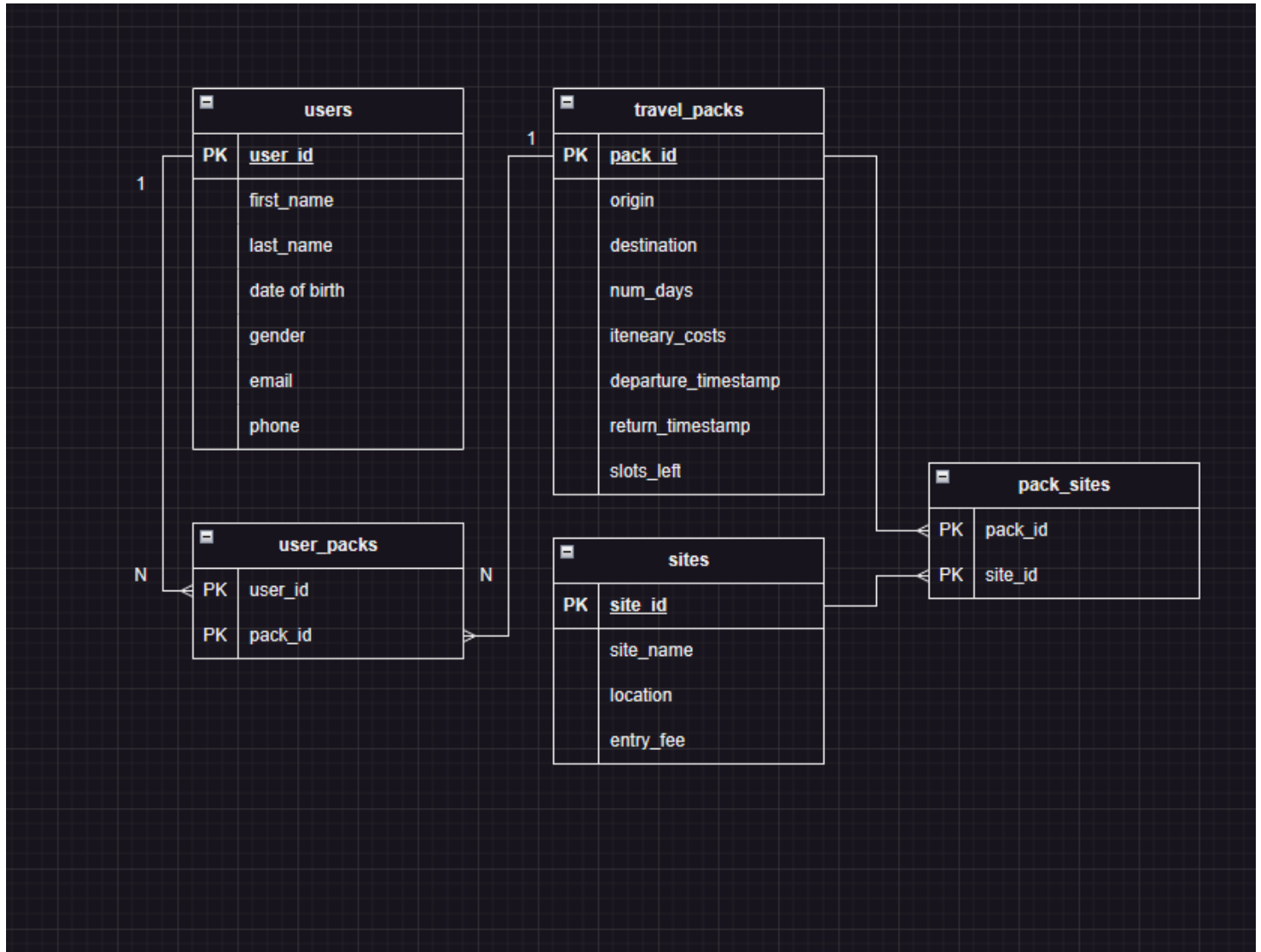
## ER Diagram





PES UNIVERSITY, BANGALORE  
Department of Computer Science and Engineering

## Relational Diagram





## Queries

### Table Creation Code

```
CREATE TABLE IF NOT EXISTS `dbms`.`users` (  
  `user_id` INT NOT NULL AUTO_INCREMENT,  
  `first_name` VARCHAR(128) NOT NULL,  
  `last_name` VARCHAR(128) NOT NULL,  
  `date_of_birth` DATE NOT NULL,  
  `gender` VARCHAR(8) NOT NULL,  
  `email` VARCHAR(128) NOT NULL,  
  `phone` VARCHAR(15) NOT NULL,  
  PRIMARY KEY(`user_id`)  
) ENGINE = InnoDB;
```

```
CREATE TABLE IF NOT EXISTS `dbms`.`travel_packs` (  
  `pack_id` INT NOT NULL AUTO_INCREMENT,  
  `origin` VARCHAR(128) NOT NULL,  
  `destination` VARCHAR(128) NOT NULL,  
  `num_days` INT NOT NULL,  
  `iteneary_costs` INT NOT NULL,  
  `departure_timestamp` DATE NOT NULL,  
  `return_timestamp` DATE NOT NULL,  
  `slots_left` INT NOT NULL DEFAULT 3,  
  PRIMARY KEY(`pack_id`)  
) ENGINE = InnoDB;
```

```
CREATE TABLE IF NOT EXISTS `dbms`.`user_packs` (  
  `pack_id` INT NOT NULL,  
  `user_id` INT NOT NULL,  
  FOREIGN KEY (pack_id) REFERENCES travel_packs(pack_id),  
  FOREIGN KEY (user_id) REFERENCES users(user_id)  
) ENGINE = InnoDB;
```

```
CREATE TABLE IF NOT EXISTS `dbms`.`sites` (  
  `site_id` INT NOT NULL AUTO_INCREMENT,  
  `site_name` VARCHAR(128) NOT NULL,  
  `address` VARCHAR(256) NOT NULL,  
  `entry_fee` FLOAT NOT NULL,  
  PRIMARY KEY (site_id)  
) ENGINE = InnoDB;
```



```
CREATE TABLE IF NOT EXISTS `dbms`.`pack_sites` (  
  `pack_id` INT NOT NULL,  
  `site_id` INT NOT NULL,  
  FOREIGN KEY (pack_id) REFERENCES travel_packs(pack_id),  
  FOREIGN KEY (site_id) REFERENCES sites(site_id),  
  PRIMARY KEY (pack_id, site_id)  
) ENGINE = InnoDB;
```

### Triggers

```
DELIMITER //  
  
CREATE TRIGGER check_seats_trigger  
BEFORE INSERT ON user_packs  
FOR EACH ROW  
BEGIN  
  DECLARE available_seats INT;  
  
  SELECT slots_left INTO available_seats FROM travel_packs WHERE pack_id  
= NEW.pack_id;  
  
  IF available_seats > 0 THEN  
    SET available_seats = available_seats - 1;  
    UPDATE travel_packs SET slots_left = available_seats WHERE pack_id  
= NEW.pack_id;  
  ELSE  
    SIGNAL SQLSTATE '45000'  
    SET MESSAGE_TEXT = 'No available seats for the selected pack.';  
  END IF;  
END //  
  
DELIMITER ;
```

The Code above reduces the number of slots in a travel pack when a new user selects that user pack. When the number of slots reaches 0, it will be throw an error



```
DELIMITER //
```

```
CREATE TRIGGER check_phone_number_trigger  
BEFORE INSERT OR UPDATE ON users  
FOR EACH ROW  
BEGIN  
    IF NEW.phone REGEXP '^(9|8|7|6)[0-9]{9}$' = 0 THEN  
        SIGNAL SQLSTATE '45000'  
        SET MESSAGE_TEXT = 'Invalid phone number format.';  
    END IF;  
END //
```

```
DELIMITER ;
```

This trigger is used to check if the phone number entered by the user is valid. It uses regex for validation of Indian mobile numbers.

## Joins

```
select pack_id, GROUP_CONCAT(CONCAT(users.user_id, " | ",  
users.first_name)) from user_packs join users on  
users.user_id=user_packs.user_id group by pack_id;
```

The above code joins travel\_packs and the users who have taken those travel packs using group by and displays it.

```
select pack_id, GROUP_CONCAT(CONCAT(sites.site_id, '|', sites.site_name))  
from pack_sites join sites on sites.site_id=pack_sites.site_id group by  
pack_id;
```

The above code joins travel\_packs and sites table to display all the sites included within each travel pack.



PES UNIVERSITY, BANGALORE  
Department of Computer Science and Engineering

```
select travel_packs.pack_id, travel_packs.destination, sites.site_id,  
sites.site_name, sites.entry_fee from pack_sites join sites on  
sites.site_id=pack_sites.site_id join travel_packs on travel_packs.pack_id  
= pack_sites.pack_id;
```

This code joins 3 tables: pack\_sites, sites and travel\_pack to display all the sites within each travel pack along with additional data such as pack\_id and destination.

## Procedures

```
DELIMITER //  
  
CREATE PROCEDURE GetPriceStats()  
BEGIN  
    DECLARE min_price DECIMAL(10, 2);  
    DECLARE max_price DECIMAL(10, 2);  
    DECLARE avg_price DECIMAL(10, 2);  
  
    SELECT MIN(iteneary_costs) INTO min_price FROM travel_packs;  
    SELECT MAX(iteneary_costs) INTO max_price FROM travel_packs;  
    SELECT AVG(iteneary_costs) INTO avg_price FROM travel_packs;  
  
    SELECT min_price AS min_price, max_price AS max_price, avg_price AS  
avg_price;  
END //  
  
DELIMITER ;
```

The above code gets the minimum, maximum and average price of travel packs





PES UNIVERSITY, BANGALORE  
Department of Computer Science and Engineering

```
DELIMITER //
```

```
CREATE PROCEDURE GetPriceRangeData(IN input_price DECIMAL(10, 2), IN  
range_value DECIMAL(10, 2))  
BEGIN  
    DECLARE min_price DECIMAL(10, 2);  
    DECLARE max_price DECIMAL(10, 2);  
  
    SET min_price = input_price - range_value;  
    SET max_price = input_price + range_value;  
  
    SELECT * FROM travel_packs  
    WHERE iteneary_costs >= min_price AND iteneary_costs <= max_price;  
END //
```

```
DELIMITER ;
```

The above code gets travel packs which are in the range of the user's travel budget. They can give a base value and +- range value.