KARSAFAR FINAL REPORT

Group Members:

Aadish Jain: 2310110004
Sane Yateesh Reddy: 2310110673
Muskan Saxena: 2310110186
Shruti Sharma: 2310110715

Code:

GitHub https://github.com/AadishJ/KarSafar

Site:

Deployed On Vercel: https:/kar-safar.vercel.app/

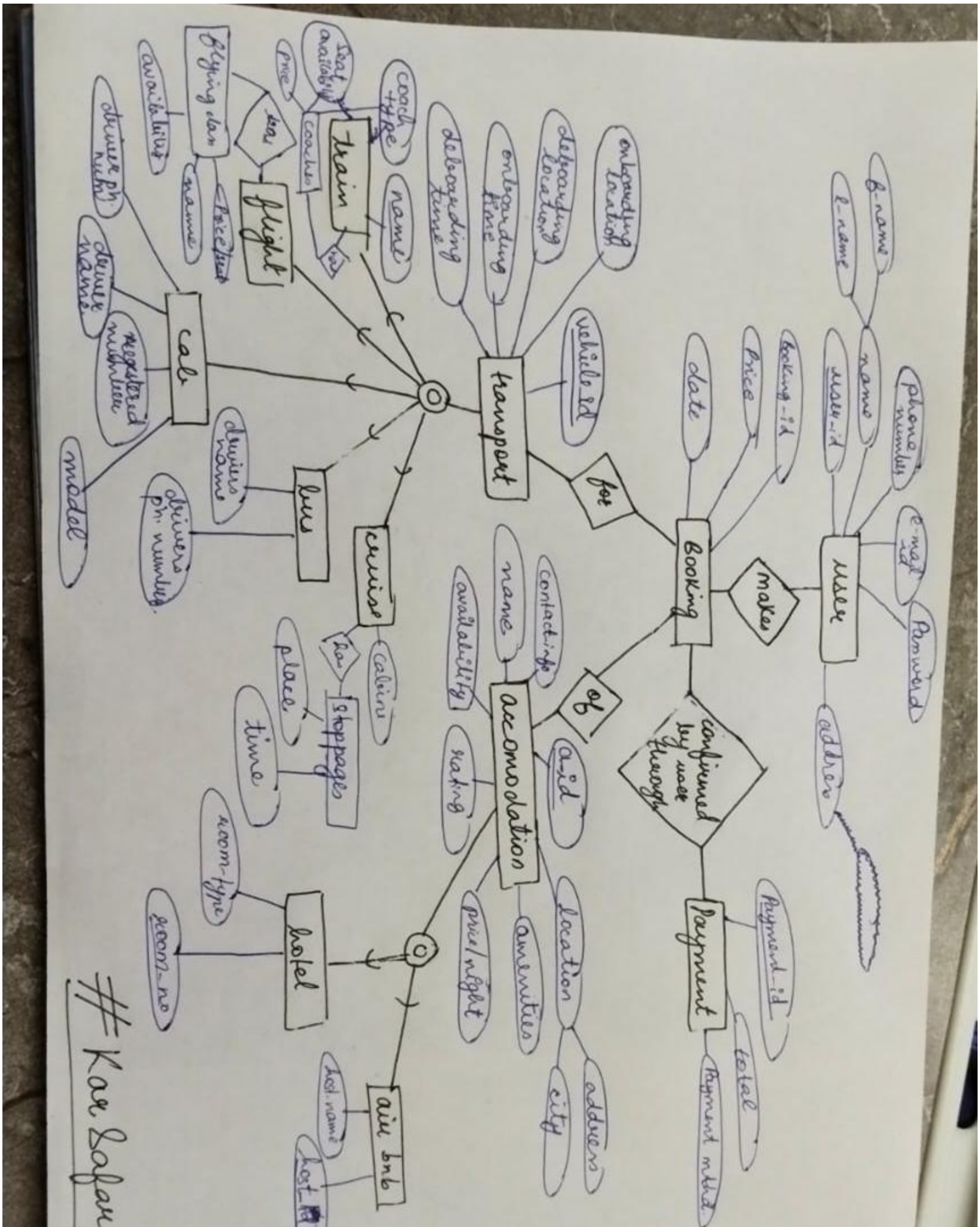
OBJECTIVE

The objective of ***KarSafar*** is to develop an ambitious, full-stack web application that serves as a comprehensive, one-stop solution for all travel-related planning and booking needs. Designed with the vision of streamlining the often-fragmented travel experience, KarSafar integrates multiple services—including flights, trains, buses, cabs, hotels, Airbnb-style vacation rentals, and even cruise bookings—into a single, seamless platform. By consolidating these offerings, the application empowers users to search, compare, and book various accommodations and transportation options with ease. The frontend, developed using React and Material UI, delivers a modern, intuitive, and responsive user interface, featuring smooth-flowing booking forms and a visually appealing layout. The backend is powered by Express.js and MySQL, ensuring robust data handling and scalability. To enhance user security and convenience, Google Authentication is incorporated for seamless and secure login. KarSafar goes beyond simple booking functionalities by offering advanced features such as trip creation, unified booking management, detailed listings with availability and amenities, and a smooth trip organization system. While tailored for the Indian market with INR-based pricing and India-specific content, the platform retains global capabilities, aiming to serve both domestic and international travelers. Overall, KarSafar aspires to redefine travel planning by providing a unified, user-friendly application that brings together all aspects of travel in one efficient digital ecosystem.

FEATURES

KarSafar is packed with a wide range of features designed to offer users a seamless and comprehensive travel planning experience. At its core, the platform supports booking across multiple categories including flights, trains, buses, cabs, hotels, Airbnb-style rentals, and even cruises—all accessible from a single interface. Each booking module is equipped with smooth, user-friendly forms and real-time availability checks, ensuring a hassle-free reservation process. The platform also includes a trip creation feature, allowing users to group multiple bookings under a single itinerary for better organization and planning. Detailed property and transport listings showcase essential information such as pricing, amenities, ratings, and availability. A unified booking management dashboard enables users to view, modify, or cancel their bookings with ease. The application is built with modern UI/UX principles using Material UI, offering a clean, responsive design that works flawlessly across devices. Additional features include Google Authentication for secure login, location-based content tailored to Indian users, pricing in Indian Rupees, and support for international travel options. Together, these features make KarSafar a robust and user-centric platform aimed at simplifying the end-to-end travel experience.

ER-MODEL


user
f-name
l-name
name
user-id
phone number
e-mail id
Password
address
make
Booking
booking-id
Price
date
for
transport
vehicle id
onboarding location
departing location
onboarding time
deboarding time
coach type
name
train
coaches
has
seat availability
price
flight
has
flying class
Price
name
availability
cab
driver ph number
driver name
registered number
model
bus
drivers name
drivers ph. number
cruise
cabins
has
stoppages
place
time
of
accomodation
contact info
name
availability
a_id
rating
location
address
city
amenities
price/night
hotel
room-type
room_no
air bnb
host name
host ph no
confirmed by user through
Payment
Payment-id
total
Payment mthd
# Kar Safar

RELATIONAL MODEL

1. **User** - userId, firstName, lastName, phoneNo, Email, Password, Street, City, State, Pincode, Country, picture
2. **Booking** - bookingId, price, status, userId, paymentId, paymentMethod, tripId, vehicleItemId, accomodationItemId
3. **VehicleItem** - vehicleItemId, vehicleId, onboardingLocation, deboardingLocation, onboardingTime, deboardingTime, coachType, seatId
4. **Seat** - vehicleId, seatId, seatNumber, name, age, gender, food
5. **VechicleItemCoach** - vehicleItemCoachId, coachType, noOfSeats
6. **Station** - vehicleId, stationName, time, stoppage
7. **Coach** - vehicleId, coachType, seatsAvailable, price
8. **Train** - vehicleId, trainName
9. **Flight** - vehicleId, flightName, photo
10. **Bus** - vehicleId, busName, driverName, driverPhoneNo, photo
11. **Cab** - vehicleId, driverName, driverPhoneNo, carModel, photo
12. **Cruise** - vehicleId, cruiseName, photo
13. **Review** - itemId, reviewId, comment, rating
14. **Pictures** - accomId, picture
15. **AccomItem** - accomItemId, accomId, checkInDate, checkoutDate, name , phoneNo, email
16. **AccomItemRoom** - accomItemId, accomId, roomType, roomNumber
17. **Hotel** - accomId, name, phoneNo, email, Street, City, State, Pincode, Country, description, breakfastInc, AC/non-AC
18. **Rooms** - accomId, roomType, roomsAvailable, pplAccomodated, roomDescr // rooom type should inc AC, non AC, suite ke b type, bunk bed
19. **Airbnb** - accomId, name, phoneNo, email, Street, landmark, City, State, Pincode, Country, description, maxAllowedGuests
20. **AirbnbAmen** - accomId, amenityType

NORMALIZED RELATIONAL SCHEMA

1. **User Management:**

a) **User** (userId PK, firstName, lastName, phoneNo, email, password, profilePicture)
b) **UserAddress** (addressId PK, userId FK, street, city, state, pinCode, country)

2. **Vehicle Management:**

a) **Vehicle** (vehicleId PK, vehicleType, status, availableSeats)
b) **Train** (vehicleId PK/FK, trainName)
c) **Flight** (vehicleId PK/FK, flightName)
d) **Bus** (vehicleId PK/FK, busName, photo)
e) **Cab** (vehicleId PK/FK, carModel, photo)
f) **Cruise** (vehicleId PK/FK, cruiseName, photo)
g) **VehicleDriver** (driverId PK, vehicleId FK, driverName, driverPhoneNo)
h) **VehicleCoach** (coachId PK, vehicleId FK, coachType, seatsAvailable, price)
i) **Stations** (stationId PK, stationName, stationType, city, state, country, latitude, longitude)
j) **Vehiclestations** (vehicleStationId PK, vehicleId FK, stationId FK, arrivalTime, departureTime, stoppage, stationOrder, UNIQUE(vehicleId, stationId), UNIQUE(vehicleId, stationOrder))
k) **Seat** (seatId PK, vehicleId FK, coachId FK, seatNumber)

3. **Accommodation Management:**

a) **Accommodation** (accomId PK, accomType, name, phoneNo, email, description)
b) **AccommodationAddress** (addressId PK, accomId FK, street, landmark, city, state, pinCode, country)
c) **AccommodationPhoto** (photoId PK, accomId FK, photoUrl)
d) **AccommodationAmenity** (amenityId PK, amenityType)
e) **AccomAmenityMap** (accomId FK, amenityId FK, PK(accomId, amenityId))
f) **Hotel** (accomId PK/FK, breakfastIncluded, acType)
g) **Airbnb** (accomId PK/FK, maxAllowedGuests)
h) **Room** (roomId PK, accomId FK, roomType, roomsAvailable, pplAccommodated, roomDescription, price)

4. **Booking Management:**

a) **Trip** (tripId PK, userId FK, name, startDate, endDate, status)
b) **VehicleBookingItem** (vehicleItemId PK, vehicleId FK, onboardingLocation, deboardingLocation, onboardingTime, deboardingTime, coachType, price, status)
c) **AccommodationBookingItem** (accomItemId PK, accomId FK, checkInDate, checkOutDate, contactName, contactPhoneNo, contactEmail, price, status)
d) **AccomBookingRoom** (bookingRoomId PK, accomItemId FK, roomId FK, roomNumber)
e) **PassengerSeat** (passengerId PK, vehicleItemId FK, seatId FK, name, age, gender, foodPreference)
f) **Booking** (bookingId PK, userId FK, tripId FK, totalPrice, status, createDate)
g) **BookingItem** (bookingItemId PK, bookingId FK, itemType, vehicleItemId FK, accomItemId FK, price)
h) **Payment** (paymentId PK, bookingId FK, amount, paid, paymentMethod, transactionId, paymentDate, status)

5. **Reviews:**

a) **Review** (reviewId PK, userId FK, itemType, itemId, rating, comment, reviewDate)

SQL QUERIES

```
CREATE TABLE users (
    userId BINARY(16) PRIMARY KEY,
    firstName VARCHAR(50) NOT NULL,
    lastName VARCHAR(50),
    phoneNo VARCHAR(15) NOT NULL,
    email VARCHAR(100) NOT NULL UNIQUE,
    password VARCHAR(256) NOT NULL,
    profilePicture VARCHAR(255)
);

CREATE TABLE useraddresses (
    addressId BINARY(16) PRIMARY KEY,
    userId BINARY(16) NOT NULL,
    street VARCHAR(100),
    city VARCHAR(50) NOT NULL,
    state VARCHAR(50),
    pinCode VARCHAR(10),
    country VARCHAR(50) NOT NULL,
    FOREIGN KEY (userId) REFERENCES users(userId) ON DELETE CASCADE
);
```

```
CREATE TABLE vehicles (
   vehicleId BINARY(16) PRIMARY KEY,
   vehicleType ENUM('train', 'flight', 'bus', 'cab', 'cruise') NOT NULL,
   status ENUM('active', 'maintenance', 'cancelled') DEFAULT 'active',
   availableSeats INT NOT NULL
);

CREATE TABLE trains (
   vehicleId BINARY(16) KEY,
   trainName VARCHAR(100) NOT NULL,
   FOREIGN KEY (vehicleId) REFERENCES vehicles(vehicleId) ON DELETE CASCADE
);

CREATE TABLE flights (
   vehicleId BINARY(16) PRIMARY KEY,
   flightName VARCHAR(100) NOT NULL,
   FOREIGN KEY (vehicleId) REFERENCES vehicles(vehicleId) ON DELETE CASCADE
);

CREATE TABLE buses (
   vehicleId BINARY(16) PRIMARY KEY,
   busName VARCHAR(100) NOT NULL,
   photo VARCHAR(255),
   FOREIGN KEY (vehicleId) REFERENCES vehicles(vehicleId) ON DELETE CASCADE
);

CREATE TABLE cabs (
   vehicleId BINARY(16) PRIMARY KEY,
   carModel VARCHAR(100) NOT NULL,
   photo VARCHAR(255),
   FOREIGN KEY (vehicleId) REFERENCES vehicles(vehicleId) ON DELETE CASCADE
);

CREATE TABLE cruises (
   vehicleId BINARY(16) PRIMARY KEY,
   cruiseName VARCHAR(100) NOT NULL,
   photo VARCHAR(255),
   FOREIGN KEY (vehicleId) REFERENCES vehicles(vehicleId) ON DELETE CASCADE
);

CREATE TABLE vehicledrivers (
   driverId BINARY(16) PRIMARY KEY,
   vehicleId BINARY(16) NOT NULL,
   driverName VARCHAR(100) NOT NULL,
   driverPhoneNo VARCHAR(15) NOT NULL,
   FOREIGN KEY (vehicleId) REFERENCES vehicles(vehicleId) ON DELETE CASCADE
);

CREATE TABLE vehiclecoaches (
   coachId VARCHAR(5) PRIMARY KEY,
   vehicleId BINARY(16) NOT NULL,
   coachType VARCHAR(50) NOT NULL,
   seatsAvailable INT NOT NULL,
   price DECIMAL(10,2) NOT NULL,
   FOREIGN KEY (vehicleId) REFERENCES vehicles(vehicleId) ON DELETE
CASCADE,
   UNIQUE KEY unique_vehicle_coach (vehicleId, coachType)
);

CREATE TABLE stations (
   stationId BINARY(16) PRIMARY KEY,
   stationName VARCHAR(100) NOT NULL,
   stationType ENUM('railway', 'airport', 'bus', 'seaport') NOT NULL,
   city VARCHAR(50) NOT NULL,
   state VARCHAR(50),
   country VARCHAR(50) NOT NULL,
   latitude DECIMAL(10,8),
   longitude DECIMAL(11,8),
   UNIQUE KEY unique_station_name_type (stationName, stationType, city)
);
```

```
CREATE TABLE vehiclestations (
   vehicleStationId BINARY(16) PRIMARY KEY,
   vehicleId BINARY(16) NOT NULL,
   stationId BINARY(16) NOT NULL,
   arrivalTime DATETIME,
   departureTime DATETIME,
   stoppage INT,
   stationOrder INT NOT NULL,
   FOREIGN KEY (vehicleId) REFERENCES vehicles(vehicleId) ON DELETE
CASCADE,
   FOREIGN KEY (stationId) REFERENCES stations(stationId) ON DELETE CASCADE,
   UNIQUE KEY unique_vehicle_station (vehicleId, stationId),
   UNIQUE KEY unique_vehicle_station_order (vehicleId, stationOrder)
);

CREATE TABLE seats (
   seatId BINARY(16) PRIMARY KEY,
   vehicleId BINARY(16) NOT NULL,
   coachId VARCHAR(5) NOT NULL,
   seatNumber VARCHAR(5) NOT NULL,
   FOREIGN KEY (vehicleId) REFERENCES vehicles(vehicleId) ON DELETE
CASCADE,
   FOREIGN KEY (coachId) REFERENCES vehiclecoaches(coachId) ON DELETE
CASCADE,
   UNIQUE KEY unique_seat (vehicleId, coachId, seatNumber)
);

CREATE TABLE accommodations (
   accomId BINARY(16) PRIMARY KEY,
   accomType ENUM('hotel', 'airbnb') NOT NULL,
   name VARCHAR(100) NOT NULL,
   phoneNo VARCHAR(15) NOT NULL,
   email VARCHAR(100),
   description TEXT
);
```

```
CREATE TABLE accommodationaddresses (
    addressId BINARY(16) PRIMARY KEY,
    accomId BINARY(16) NOT NULL,
    street VARCHAR(100) NOT NULL,
    landmark VARCHAR(100),
    city VARCHAR(50) NOT NULL,
    state VARCHAR(50),
    pinCode VARCHAR(10),
    country VARCHAR(50) NOT NULL,
    FOREIGN KEY (accomId) REFERENCES accommodations(accomId) ON DELETE
CASCADE
);

CREATE TABLE accommodationphotos (
    photoId BINARY(16) PRIMARY KEY,
    accomId BINARY(16) NOT NULL,
    photoUrl VARCHAR(255) NOT NULL,
    FOREIGN KEY (accomId) REFERENCES accommodations(accomId) ON DELETE
CASCADE
);

CREATE TABLE accommodationamenities (
    amenityId BINARY(16) PRIMARY KEY,
    amenityType VARCHAR(50) NOT NULL UNIQUE
);

CREATE TABLE accomamenitymap (
    accomId BINARY(16) NOT NULL,
    amenityId BINARY(16) NOT NULL,
    PRIMARY KEY (accomId, amenityId),
    FOREIGN KEY (accomId) REFERENCES accommodations(accomId) ON DELETE
CASCADE,
    FOREIGN KEY (amenityId) REFERENCES accommodationamenities(amenityId) ON
DELETE CASCADE
);
```

```
CREATE TABLE hotels (
   accomId BINARY(16) PRIMARY KEY,
   breakfastIncluded BOOLEAN DEFAULT FALSE,
   acType ENUM('AC', 'NON-AC', 'BOTH') NOT NULL,
   FOREIGN KEY (accomId) REFERENCES accommodations(accomId) ON DELETE
CASCADE
);

CREATE TABLE airbnbs (
   accomId BINARY(16) PRIMARY KEY,
   maxAllowedGuests INT NOT NULL,
   FOREIGN KEY (accomId) REFERENCES accommodations(accomId) ON DELETE
CASCADE
);

CREATE TABLE rooms (
   roomId BINARY(16) PRIMARY KEY,
   accomId BINARY(16) NOT NULL,
   roomType VARCHAR(50) NOT NULL,
   roomsAvailable INT NOT NULL,
   pplAccommodated INT NOT NULL,
   roomDescription TEXT,
   price DECIMAL(10,2) NOT NULL,
   FOREIGN KEY (accomId) REFERENCES accommodations(accomId) ON DELETE
CASCADE
);


CREATE TABLE trips (
   tripId BINARY(16) PRIMARY KEY,
   userId BINARY(16) NOT NULL,
   name VARCHAR(100) NOT NULL,
   startDate DATE NOT NULL,
   endDate DATE NOT NULL,
   status ENUM('planning', 'booked', 'ongoing', 'completed', 'cancelled') DEFAULT
'planning',
   FOREIGN KEY (userId) REFERENCES users(userId) ON DELETE CASCADE
);
```

```
CREATE TABLE vehiclebookingitems (
   vehicleItemId BINARY(16) PRIMARY KEY,
   vehicleId BINARY(16) NOT NULL,
   onboardingLocation VARCHAR(100) NOT NULL,
   deboardingLocation VARCHAR(100) NOT NULL,
   onboardingTime DATETIME NOT NULL,
   deboardingTime DATETIME NOT NULL,
   coachType VARCHAR(50),
   price DECIMAL(10,2) NOT NULL,
   status ENUM('confirmed', 'pending', 'cancelled') DEFAULT 'pending',
   FOREIGN KEY (vehicleId) REFERENCES vehicles(vehicleId) ON DELETE CASCADE
);

CREATE TABLE accommodationbookingitems (
   accomItemId BINARY(16) PRIMARY KEY,
   accomId BINARY(16) NOT NULL,
   checkInDate DATE NOT NULL,
   checkOutDate DATE NOT NULL,
   contactName VARCHAR(100) NOT NULL,
   contactPhoneNo VARCHAR(15) NOT NULL,
   contactEmail VARCHAR(100),
   price DECIMAL(10,2) NOT NULL,
   status ENUM('confirmed', 'pending', 'cancelled') DEFAULT 'pending',
   FOREIGN KEY (accomId) REFERENCES accommodations(accomId) ON DELETE
CASCADE
);

CREATE TABLE accombookingrooms (
   bookingRoomId BINARY(16) PRIMARY KEY,
   accomItemId BINARY(16) NOT NULL,
   roomId BINARY(16) NOT NULL,
   roomNumber VARCHAR(20),
   FOREIGN KEY (accomItemId) REFERENCES
accommodationbookingitems(accomItemId) ON DELETE CASCADE,
   FOREIGN KEY (roomId) REFERENCES rooms(roomId) ON DELETE CASCADE
);
```

```
CREATE TABLE passengerseats (
   passengerId BINARY(16) PRIMARY KEY,
   vehicleItemId BINARY(16) NOT NULL,
   seatId BINARY(16) NOT NULL,
   name VARCHAR(100) NOT NULL,
   age INT NOT NULL,
   gender ENUM('male', 'female', 'other') NOT NULL,
   foodPreference ENUM('veg', 'non-veg', 'vegan', 'none') DEFAULT 'none',
   FOREIGN KEY (vehicleItemId) REFERENCES vehiclebookingitems(vehicleItemId)
ON DELETE CASCADE,
   FOREIGN KEY (seatId) REFERENCES seats(seatId) ON DELETE CASCADE,
   UNIQUE KEY unique_vehicle_seat (vehicleItemId, seatId)
);

CREATE TABLE bookings (
   bookingId BINARY(16) PRIMARY KEY,
   userId BINARY(16) NOT NULL,
   tripId BINARY(16),
   totalPrice DECIMAL(10,2) NOT NULL,
   status ENUM('confirmed', 'pending', 'cancelled') DEFAULT 'pending',
   createDate TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
   FOREIGN KEY (userId) REFERENCES users(userId) ON DELETE CASCADE,
   FOREIGN KEY (tripId) REFERENCES trips(tripId) ON DELETE SET NULL
);
CREATE TABLE bookingitems (
   bookingItemId BINARY(16) PRIMARY KEY,
   bookingId BINARY(16) NOT NULL,
   itemType ENUM('vehicle', 'accommodation') NOT NULL,
   vehicleItemId BINARY(16),
   accomItemId BINARY(16),
   price DECIMAL(10,2) NOT NULL,
   FOREIGN KEY (bookingId) REFERENCES bookings(bookingId) ON DELETE
CASCADE,
   FOREIGN KEY (vehicleItemId) REFERENCES vehiclebookingitems(vehicleItemId)
ON DELETE SET NULL,
   FOREIGN KEY (accomItemId) REFERENCES
accommodationbookingitems(accomItemId) ON DELETE SET NULL
);
```

```
DELIMITER //

CREATE TRIGGER before_insert_bookingItems
BEFORE INSERT ON bookingitems
FOR EACH ROW
BEGIN
    -- Ensure only one of vehicleItemId or accomItemId is set based on itemType
    IF (NEW.itemType = 'vehicle' AND (NEW.vehicleItemId IS NULL OR NEW.accomItemId IS NOT NULL)) OR
       (NEW.itemType = 'accommodation' AND (NEW.accomItemId IS NULL OR NEW.vehicleItemId IS NOT NULL)) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Invalid itemType: vehicleItemId and accomItemId must match itemType';
    END IF;
END //

CREATE TRIGGER before_update_bookingItems
BEFORE UPDATE ON bookingitems
FOR EACH ROW
BEGIN
    -- Ensure only one of vehicleItemId or accomItemId is set based on itemType
    IF (NEW.itemType = 'vehicle' AND (NEW.vehicleItemId IS NULL OR NEW.accomItemId IS NOT NULL)) OR
       (NEW.itemType = 'accommodation' AND (NEW.accomItemId IS NULL OR NEW.vehicleItemId IS NOT NULL)) THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Invalid itemType: vehicleItemId and accomItemId must match itemType';
    END IF;
END //
DELIMITER ;
```

```
CREATE TABLE payments (
   paymentId BINARY(16) PRIMARY KEY,
   bookingId BINARY(16) NOT NULL,
   amount DECIMAL(10,2) NOT NULL,
   paid BOOLEAN DEFAULT FALSE,
   paymentMethod VARCHAR(50),
   transactionId VARCHAR(100),
   paymentDate TIMESTAMP,
   status ENUM('pending', 'completed', 'failed', 'refunded') DEFAULT 'pending',
   FOREIGN KEY (bookingId) REFERENCES bookings(bookingId) ON DELETE
CASCADE
);
```

```
CREATE TABLE reviews (
   reviewId BINARY(16) PRIMARY KEY,
   userId BINARY(16) NOT NULL,
   itemType ENUM('vehicle', 'accommodation', 'trip') NOT NULL,
   itemId BINARY(16) NOT NULL,
   rating DECIMAL(3,2) NOT NULL CHECK (rating BETWEEN 0 AND 5),
   comment TEXT,
   reviewDate TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
   FOREIGN KEY (userId) REFERENCES users(userId) ON DELETE CASCADE,
   UNIQUE KEY unique_user_review (userId, itemType, itemId)
);
```

CODE

GitHub github.com/AadishJ/KarSafar

SITE - https://kar-safar.vercel.app/

REFRENCES

Frontend Technologies

React Documentation

Material UI Documentation

Material Icons Reference

React Router Documentation

date-fns Documentation

Axios Documentation

Backend Technologies

MySQL Documentation

SQL Syntax Reference

Database Design References

User Management

Auth0 User Authentication Guide

UI/UX Design References

Booking.com Design System

Airbnb Design

Component Design

Material Design Guidelines

Travel UI Component Libraries

React Icons

Tailwind Documentation

API Design References

REST API Design Guidelines

HTTP Status Code Reference

Project Management References

React Application Architecture