Admin Dashboard Schema

Login

Admins Table:

admin_id (PK)

username

email

password_hash

remember_token

Explanation:

- **admin_id**: Primary key for the admin.
- **username**: Admin's username.
- **email**: Admin's email address.
- **password_hash**: Hashed password for security.
- **remember_token**: Token to remember the admin's session

Operations:

Admin Login: Admin will enter their username and password. If they check "Remember this device", a token will be generated and stored in the database for future authentication.

Admin Logout: Admin's remember token will be cleared from the database upon logout.

Admin Login:

```
SELECT * FROM Admins WHERE username = 'admin_username' AND password_hash = 'hashed_password';
```

```
UPDATE Admins SET remember_token = 'generated_token' WHERE admin_id = <admin_id>;
```

Admin Logout:

```
UPDATE Admins SET remember_token = NULL WHERE admin_id = <admin_id>;
```

OTP in the Admins table

Admins Table:

admin_id (PK)

username

email

password_hash

remember_token

otp_code

otp_expiry

Steps to implement OTP sending:

Generate a random OTP.

Store the OTP and its expiry time in the Admins table.

Send the OTP to the admin's registered email or phone number.

Verify the OTP when the admin submits it.

Generating and Storing OTP:

```
UPDATE Admins
SET otp_code = 'generated_otp', otp_expiry = DATE_ADD(NOW(), INTERVAL 10 MINUTE)
WHERE admin_id = <admin_id>;
```

Sending OTP to Admin:

This step involves sending the OTP through email or SMS. If you're using email, you can send it using a library like nodemailer in Node.js. If it's SMS, you might use a service like Twilio.

3. Verifying OTP:

```
SELECT * FROM Admins WHERE admin_id = <admin_id> AND otp_code = 'entered_otp' AND otp_expiry > NOW();
```

Invite User

- Admin Form: Create a form with fields for Name, Email, Phone Number, and Message.

- Validate Input: Ensure the input is valid before sending the invitation.
- Send Invitation: Once validated, send the invitation through email or SMS.

Sending Invitation:

```
INSERT INTO Invitations (name, email, phone, message, sent_date)
VALUES ('John Doe', 'john@example.com', '1234567890', 'Join our platform!', NOW());
```

Invite Driver

Invitations Table:

invitation_id (PK)
driver_name
driver_email
phone_number
message
invitation_status
created_at

Steps to Invite a Driver:

- Admin fills out the invitation form.
- Admin sends the invitation.
- Optionally, log the invitation in the database

Sending Invitation:

```
INSERT INTO Invitations (driver_name, driver_email, phone_number, message,
invitation_status, created_at)
VALUES ('driver_name', 'driver_email@example.com', 'phone_number', 'message',
'Pending', NOW());
```

Logging Invitation:

```
SELECT * FROM Invitations WHERE invitation_id = <invitation_id>;
```

Invite Staff

Staff Table:

staff_id (PK)
name
email
phone_number

Steps to Implement:

- Admin enters staff details.
- Admin sends an invitation.

- Staff receives the invitation.

Inserting Staff Details:

```
INSERT INTO Staff (name, email, phone_number)
VALUES ('John Doe', 'john@example.com', '+1234567890');
```