**SplitBuuddy – Real-Time Group Expense Sharing**

**Group Name : Nidhi Patel**

**Neel Udhnawala**

**Manan Kansara**

**Date : 12/07/2025**

**React : -**

* + Textbox & Radio Button using bootstrap
  + Learning Routing
  + Use navigation url

**Date : 19/07/2025**

**Node : -**

* Create an api for crud operations.
* Link with react js.

**React : -**

* Api url link with node and express js.

**Date : 28/07/25 to 02/08/25**

**Node + Express : -**

• Concept of foreign key

• Create an API of Brand and Product for crud operation

• Both API of Brand and Product in a same page

• Multiple File/Image Upload Concept  
• Session Data Storage at both side Admin and User

• Payment gateway using Razorpay

**React : -**

• Create two pages for Brand and Product display

* **View**

List of Brand

List of Product

* **Insert** Insert Brand / Insert Product

SearchBar

Add multiple image at a time

* **Edit**

Edit Records in both Brand and Products

Edit file or image and display image or file

* **Delete**

Delete Brand if Product is Empty

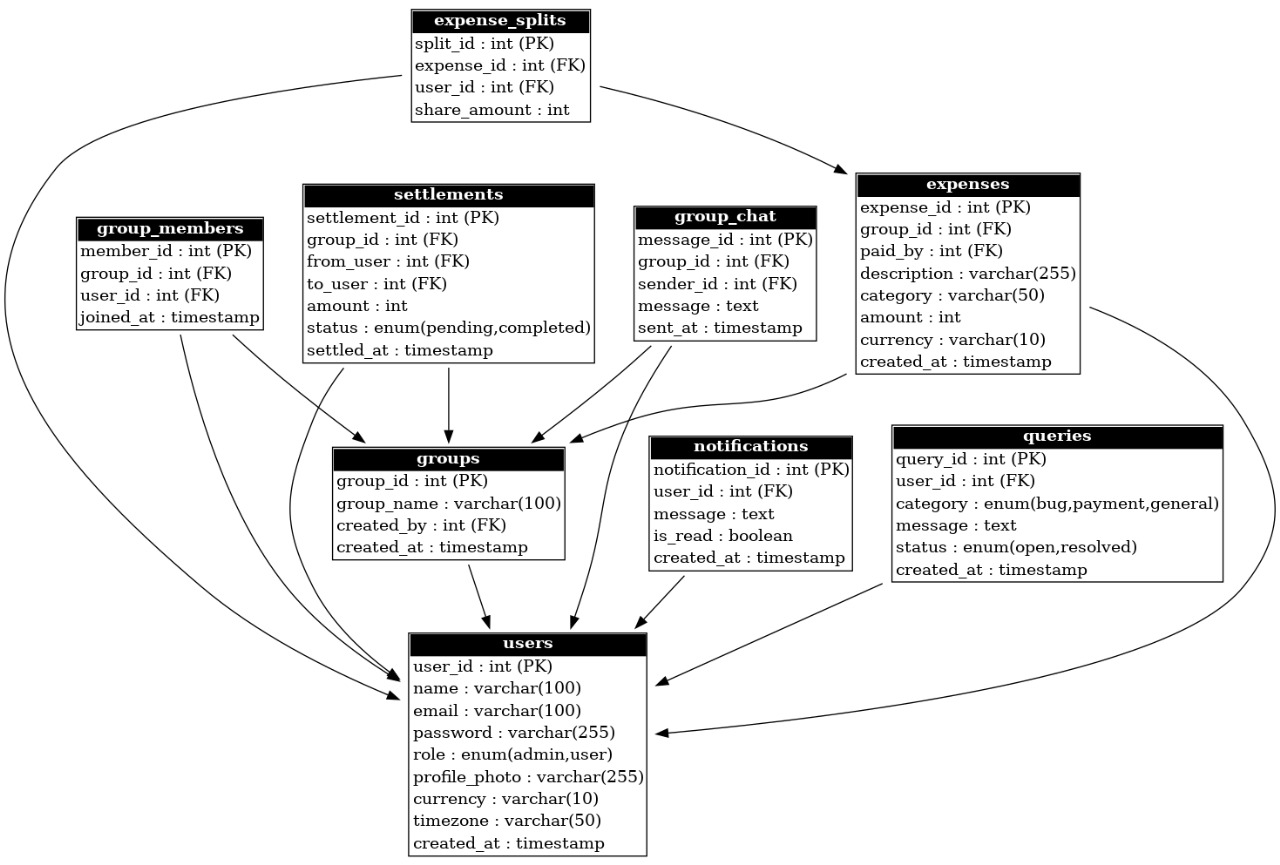
Product delete base on Product\_Id

• Set Session on the Registration and Login

and Session Expire on the Logout ButtonClick

**Date : 12/08/25**

**FINAL\_DATABASE:-**

 **Database Name**

CREATE DATABASE splitbuddy;

USE splitbuddy;

**1. Users Table**

Stores both normal users and admins (role-based access).

CREATE TABLE users (

user\_id INT AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(100) NOT NULL,

email VARCHAR(100) UNIQUE NOT NULL,

password VARCHAR(255) NOT NULL,

role ENUM('admin', 'user') DEFAULT 'user',

profile\_photo VARCHAR(255),

currency VARCHAR(10) DEFAULT 'INR',

timezone VARCHAR(50) DEFAULT 'Asia/Kolkata',

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

**2. Groups Table**

CREATE TABLE groups (

group\_id INT AUTO\_INCREMENT PRIMARY KEY,

group\_name VARCHAR(100) NOT NULL,

created\_by INT NOT NULL,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (created\_by) REFERENCES users(user\_id) ON DELETE CASCADE

);

**3. Group Members Table**

CREATE TABLE group\_members (

member\_id INT AUTO\_INCREMENT PRIMARY KEY,

group\_id INT NOT NULL,

user\_id INT NOT NULL,

joined\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (group\_id) REFERENCES groups(group\_id) ON DELETE CASCADE,

FOREIGN KEY (user\_id) REFERENCES users(user\_id) ON DELETE CASCADE

);

**4. Expenses Table**

CREATE TABLE expenses (

expense\_id INT AUTO\_INCREMENT PRIMARY KEY,

group\_id INT NOT NULL,

paid\_by INT NOT NULL,

description VARCHAR(255) NOT NULL,

category VARCHAR(50),

amount INT NOT NULL,

currency VARCHAR(10) DEFAULT 'INR',

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (group\_id) REFERENCES groups(group\_id) ON DELETE CASCADE,

FOREIGN KEY (paid\_by) REFERENCES users(user\_id) ON DELETE CASCADE

);

**5. Expense Splits Table**

CREATE TABLE expense\_splits (

split\_id INT AUTO\_INCREMENT PRIMARY KEY,

expense\_id INT NOT NULL,

user\_id INT NOT NULL,

share\_amount INT NOT NULL,

FOREIGN KEY (expense\_id) REFERENCES expenses(expense\_id) ON DELETE CASCADE,

FOREIGN KEY (user\_id) REFERENCES users(user\_id) ON DELETE CASCADE

);

**6. Settlements Table**

CREATE TABLE settlements (

settlement\_id INT AUTO\_INCREMENT PRIMARY KEY,

group\_id INT NOT NULL,

from\_user INT NOT NULL,

to\_user INT NOT NULL,

amount INT NOT NULL,

status ENUM('pending', 'completed') DEFAULT 'pending',

settled\_at TIMESTAMP NULL,

FOREIGN KEY (group\_id) REFERENCES groups(group\_id) ON DELETE CASCADE,

FOREIGN KEY (from\_user) REFERENCES users(user\_id),

FOREIGN KEY (to\_user) REFERENCES users(user\_id)

);

**7. Notifications Table**

CREATE TABLE notifications (

notification\_id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT NOT NULL,

message TEXT NOT NULL,

is\_read BOOLEAN DEFAULT FALSE,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (user\_id) REFERENCES users(user\_id) ON DELETE CASCADE

);

**8. Queries Table (Admin Support)**

CREATE TABLE queries (

query\_id INT AUTO\_INCREMENT PRIMARY KEY,

user\_id INT NOT NULL,

category ENUM('bug', 'payment', 'general') DEFAULT 'general',

message TEXT NOT NULL,

status ENUM('open', 'resolved') DEFAULT 'open',

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (user\_id) REFERENCES users(user\_id) ON DELETE CASCADE

);

**9. Chat Table (Group Chat Feature)**

CREATE TABLE group\_chat (

message\_id INT AUTO\_INCREMENT PRIMARY KEY,

group\_id INT NOT NULL,

sender\_id INT NOT NULL,

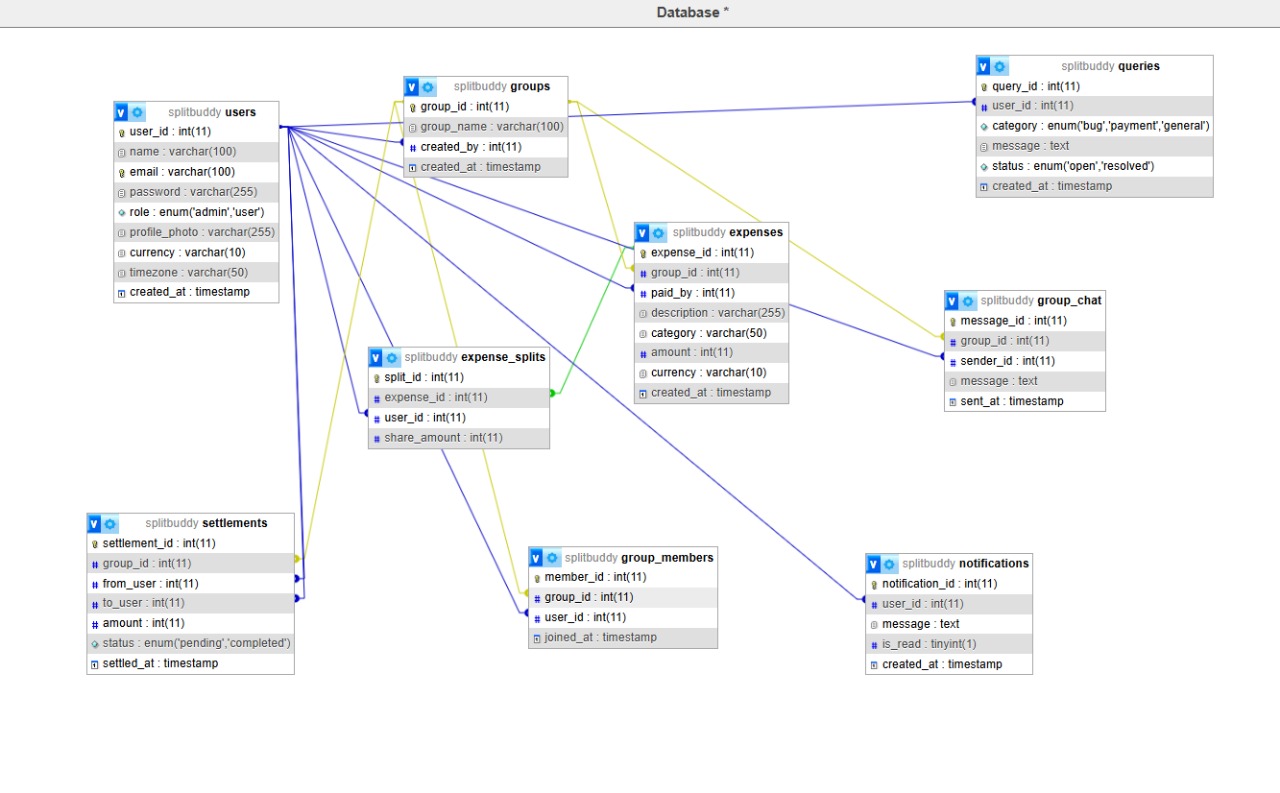
message TEXT NOT NULL,

sent\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (group\_id) REFERENCES groups(group\_id) ON DELETE CASCADE,

FOREIGN KEY (sender\_id) REFERENCES users(user\_id) ON DELETE CASCADE

);



**Date : 12/08/25 to 16/08/25**

**AdminSide**

* Login and Logout functionality
* Admin Profile
* On a Dachboard :

Sidebar: -Overview

-Active Users

-User Management

-Queries

-Group Monitoring

-Export Data

**Date : 21/08/25**

* UserSide
* Registration,Login,Logout functionality
* HomePage
* User Profile

**Date : 22/08/25 to 06/09/25**

* UserSide
* Registration,Login,Logout functionality
* HomePage
* Groups with GroupDetails
* Group Leader can Create Group , Add Members and Remove Group Members
* Add Expenses
* Show Balance( + Total Balance)

**Date : 06/09/25 to 15/09/25**

* AdminSide 80% done
* UserSide 80% done
* Registration,Login,Logout functionality
* HomePage
* Groups with GroupDetails
* Group Leader can Create Group , Add Members and Remove Group Members
* Add Expenses
* Show Balance( + Total Balance)
* Split Balance between Group Members
* Settlements between Group Members
* Update Settlement status Pending to Completed