Congratulations on learning a new web technology - **DBMS** :)

Now that you have coded the web pages of the PG-Life web application, it is now time to move to the next step. In this assignment, your task is to design an ideal database design for the PG-Life web application i.e. you will design and create a database.

What do you need to do?

1) Go through the web pages - home page, property list page, property detail page, and dashboard page. Think about all the functionalities you would build and then try to figure out different entities you would like to store in the database.
2) List down those entities and think about the relationships between each pair of them.
3) Once you know the relationships, you can then identify where the foreign keys will go. And, then you can create the tables accordingly.

Getting Started

To help you get started with the assignment, I have listed down the entities that can be stored in a database.

Entities:

- users
- cities
- properties
- amenities
- testimonials

Also, for a few entities - I have listed their relationship below. Your task is to first identify the relationships between the entities and then identify the type of database relationship between them (this will help you to add the foreign key at appropriate tables).

Relationships:

- users - cities: No, they are not related.
- users - properties: Yes, users can mark favorite properties.
- users - amenities: No, they are not related.
- users - testimonials: Yes, users give testimonials.
- cities - properties: Yes, cities contain properties.
- cities - amenities: No, they are not related.
- cities - testimonials: No, they are not related.
- properties - amenities: Yes, properties have amenities.
- properties - testimonials: Yes, properties give testimonials.
- amenities - testimonials: No, they are not related.

users - properties:

Many to Many

users - testimonials:

One to Many

cities - properties:

One to Many

properties - amenities:

Many to Many

properties - testimonials:

One to Many

Foreign keys:

S.No.	Relationship Type	Foreign Key
1	users - properties - Many to Many	interested_users_properties (intermediate table): user_id, property_id (foreign keys)
2	users - testimonials - One to Many	user_id, property_id (foreign keys)
3	cities - properties - One to Many	property_id (foreign key)
4	properties - amenities - Many to Many	properties_amenities (intermediate table): property_id, amenity_id (foreign keys)
5	properties - testimonials - One to Many	user_id, property_id (foreign keys)

Final Tables: You can find the tables and the SQL queries you can use to create these tables below.

Note that the tables should be created in exactly the same order that I have listed as we are using foreign keys to link these tables.

- users

```
CREATE TABLE users (
  id INT NOT NULL AUTO_INCREMENT,
  email VARCHAR(100) NOT NULL,
  password VARCHAR(100) NOT NULL,
  full_name VARCHAR(100) NOT NULL,
  phone VARCHAR(20) NOT NULL,
  gender ENUM('male', 'female') NOT NULL,
  college_name VARCHAR(100) NOT NULL,
  PRIMARY KEY(id)
);
```

- cities

```
CREATE TABLE cities (
  id INT NOT NULL AUTO_INCREMENT,
  name VARCHAR(100) NOT NULL,
  PRIMARY KEY(id)
);
```

- properties: city_id

```
CREATE TABLE properties (
```

```
  id INT NOT NULL AUTO_INCREMENT,
  city_id INT NOT NULL,
  name VARCHAR(100) NOT NULL,
  address VARCHAR(255) NOT NULL,
  description LONGTEXT NOT NULL,
  gender ENUM('male', 'female', 'unisex') NOT NULL,
  rent INT NOT NULL,
  rating_clean FLOAT(2,1) NOT NULL,
  rating_food FLOAT(2,1) NOT NULL,
  rating_safety FLOAT(2,1) NOT NULL,
  PRIMARY KEY(id),
  FOREIGN KEY(city_id) REFERENCES cities(id)
);
```

- amenities

```
CREATE TABLE amenities (
  id INT NOT NULL AUTO_INCREMENT,
  name VARCHAR(100) NOT NULL,
  type VARCHAR(100) NOT NULL,
  icon VARCHAR(30) NOT NULL,
  PRIMARY KEY(id)
);
```

- testimonials: user_id, property_id

```
CREATE TABLE properties_amenities (
  id INT NOT NULL AUTO_INCREMENT,
  property_id INT NOT NULL,
  amenity_id INT NOT NULL,
  PRIMARY KEY(id),
  FOREIGN KEY(property_id) REFERENCES properties(id),
  FOREIGN KEY(amenity_id) REFERENCES amenities(id)
);
```

- interested_users_properties: user_id, property_id

```
CREATE TABLE testimonials (
  id INT NOT NULL AUTO_INCREMENT,
  property_id INT NOT NULL,
  user_name VARCHAR(100) NOT NULL,
  content LONGTEXT NOT NULL,
  PRIMARY KEY(id),
  FOREIGN KEY(property_id) REFERENCES properties(id)
);
```

- properties_amenities: property_id, amenity_id

```
CREATE TABLE interested_users_properties (
  id INT NOT NULL AUTO_INCREMENT,
```

```
  user_id INT NOT NULL,
  property_id INT NOT NULL,
  PRIMARY KEY(id),
  FOREIGN KEY(user_id) REFERENCES users(id),
  FOREIGN KEY(property_id) REFERENCES properties(id)
);
```

Note: Once tables are created, add the dummy data into the tables from dummy_data.sql