House Renting Platform – Database Design Documentation

This documentation provides an exact database design based on the Django models found in accounts/models.py and listing/models.py. No assumptions or structural modifications have been made. This document reflects the current codebase exactly.

1. Introduction

The House Renting Platform allows users to register and list properties for rent. This document presents the database design directly derived from the actual Django models provided by the development team.

2. Profile Model (accounts/models.py)

Field	Type	Description
id	AutoField (PK)	Primary key
user	OneToOneField to User	Links to Django user model
avatar	ImageField	Optional profile picture

3. ListingModel (listing/models.py)

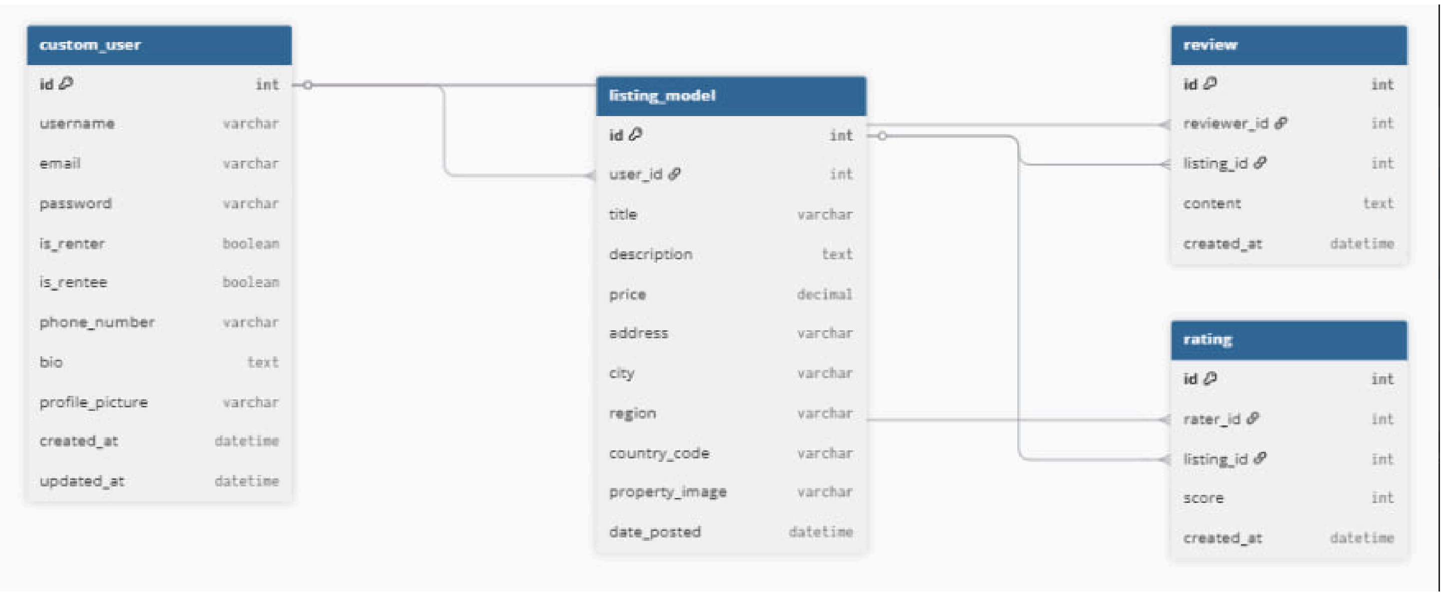
Field	Type	Description
id	AutoField (PK)	Primary key
title	CharField(255)	Property title
description	TextField	Full property description
city_address	CharField(255)	City-level address
zipcode	IntegerField	Postal/zip code
country	CharField(3)	Country code (e.g., ETH, KEN)
province	CharField	Province/region
ownership_documents	CharField	Type of ownership document
location	CharField	General location info
price	DecimalField(10,2)	Listing price
bank_debt	CharField	Indicates bank debt status
property_type	CharField	Type: Apartment, Villa, etc.
property_status	CharField	For Sale / Rent
property_condition	CharField	Condition of the property
property_size	DecimalField	Size in square meters
property_landarea	DecimalField	Total land area
furnished	CharField	Whether it’s furnished
property_rooms	IntegerField	Total rooms
bedrooms	IntegerField	Bedroom count

bathrooms	IntegerField	Bathroom count
parking	CharField	Parking availability
kitchens	IntegerField	Number of kitchens
balcony	IntegerField	Number of balconies
property_image	ImageField	Main image of the property
link	URLField	External listing link
latitude	DecimalField(9,6)	Location latitude
longitude	DecimalField(9,6)	Location longitude
date_created	DateTimeField	Timestamp when listing was added

4. Relationships

- Profile ⇔ User: One-to-One (each profile is linked to a single user)
- ListingModel: linked to a user and review and rating tables.

5. SQL Schema



custom_user	
id	int
username	varchar
email	varchar
password	varchar
is_renter	boolean
is_rentee	boolean
phone_number	varchar
bio	text
profile_picture	varchar
created_at	datetime
updated_at	datetime

listing_model	
id	int
user_id	int
title	varchar
description	text
price	decimal
address	varchar
city	varchar
region	varchar
country_code	varchar
property_image	varchar
date_posted	datetime

review	
id	int
reviewer_id	int
listing_id	int
content	text
created_at	datetime

rating	
id	int
rater_id	int
listing_id	int
score	int
created_at	datetime

7. Conclusion

This document accurately reflects the models defined in your project’s accounts/models.py and listing/models.py files. The structure, fields, and data types match your code exactly. Optional improvements are provided separately and should only be applied with team consensus.