PROJECT PHASE - 3
Team: Curious_hers
Yeduru Rupasree - 2020101097
Mahesh Sathvika - 2020101087
Sudi Ananya - 2020101075

Explanation:

ER model to relational data model:

1. As cards, review,order_details are weak entities, primary keys of entities on which they are dependent are added to their tables respectively.
2. As payment-order is a 1:1 binary relationship primary key of ORDER, 'order ID' is added as a foreign key to payment.
3. As product-supplier and order-shipper are many-to-one relationships ,sellerId is added as foreign key to product and shipperId is added as foreign key to order.
4. For multivalued attributes like 'phone no','Available Sizes', new relations named Phone_no which contains phone_no ,userID(as foreign key) as attributes and 'Available_sizes' contains size,productcode(as foreign key) as attributes.

1 NF:
The above model is already in 1NF as all attributes are atomic.

2NF:
A relation schema is in 2NF if every non-prime attribute A in R is fully functionally dependent on every key of R.
We have following relations in our relational model:
→ order_details:UserID+OrderID+ProductCode→ Quantity,size
Nonprime attributes in Order details are fully functionally dependent on UserID+OrderID+ProductCode
→Review: ProductCode+UserID→Rating,Quality,text etc.
Every non prime attribute in 'Review' is fully functionally dependent on Productcode+UserID
→ Cards: UserID+ CardNo → CardType, expiry month, expiry year.
Every non prime attribute of Cards is fully functionally dependent on UserID+CardNo
In the above-mentioned relations which contain composite key, we find that all non prime attributes are fully functionally dependent on the key. Hence, these don't violate 2NF rules.

→User: UserID → All attributes in relation table 'USER'
→ UserID is a key attribute.
→ Product: ProductCode→ Name, ProductAvailability, …..all attributes in Product Relation Where Product Code is primary key
→ Supplier: SellerID→ all attributes in Supplier table where SellerId is primary key.
→ Payment: PaymentID→ all attributes in payment relation where PaymentId is a key attribute.
Similarly, we've almost all primary keys as simple attributes. So, we'll not have the case where a non-prime attribute is partially dependent on a key. there are no partial functional dependencies on keys. Hence, the same relational model is in 2NF.

3rd Normal Form: A relation schema is in 3NF if for all non trivial dependencies in F+ are of the form X→ A with either:
a). X is a superkey
b). A is a prime attribute
Here, we've all functional dependencies which have X as a primary key.
Example:1. UserId+ProductCode+OrderId → Pricetobepaid (in Orderdetails)
Here, UserId+ProductCode+OrderId is a primary Key (also a super key).
2.ProductCode+name → stock (in Product)
Here ProductCode+name is a superkey.
Similarly, no matter which functional dependency is picked which is of form X->A, we find X as superkey.
Hence, the same relational model is in 3rd Normal form.