SCENARIO ONE

Two users try to book the same hall and the same time slot simultaneously.
Atomicity: Either one of the transactions succeeds and the other is rejected or both transactions are rolled back.
Consistency: One of the users is successful in booking the hall and its status is turned to unavailable hence the other user is not able to book it and double bookings are rejected.
Isolation: User one's booking is committed first and then user two proceeds and is rejected because the booking is unavailable.
Durability: Once a user books the hall, the change is permanent and even if the system crashes, the booking is persistent.
Concurrency Strategy: It is very rare for two users to book the same hall and time slot simultaneously so we will only check for conflict while committing the transaction.
Isolation Level Justification: The system executes the transactions in a sequential order in order to make it impossible for two users to book the same hall and slot at the exact same time.

SCENARIO TWO

User tries to cancel/refund their booking.

Atomicity: If at any point the transaction fails then it is completely rolled back and neither the refund nor the cancellation is processed and the booking remains intact.

Consistency: If the booking is successfully cancelled then the correct refund amount is calculated according to our logic.

Isolation: Other users would not be able to book the hall until the cancellation process is completed and changes are committed.

Durability: Once the cancellation process is completed, it cannot be undone and the hall is made available for booking.

Concurrency Strategy: When a user is processing a cancellation request, all other transactions related to booking are not allowed to interfere with it.

Isolation Level Justification: Once the user cancels their booking successfully, it is then the hall is made available for others to prevent conflicts and double bookings.

SCENARIO THREE

Admin cancels a booking.

Atomicity: Either the booking is completely cancelled or it remains totally fine.

Consistency: When the booking is cancelled, the refund amount is calculated upon our defined rules.

Isolation: During the process, no other transaction is allowed to interfere with this.

Durability: Once the booking is cancelled, the slot is made available again for bookings.

Concurrency Strategy: We use 2 Phase Locking in which the system locks all associated records at the start of the process and the locks are released once the cancellation process is complete.

Isolation Level Justification: All other transactions are not allowed to interfere in order to prevent multiple admins from cancelling the same booking.