

# University Room Booking - Database Configuration

## 1. Database Overview

Configuration	Details
Database System	MongoDB
ODM / Framework	Mongoose (Node.js)
Architecture	Document-Oriented (Collections)

## 2. Collection Schemas: A. Users Collection

Field	Type	Constraints	Description
_id	ObjectId	Auto-generated	Unique ID
username	String	Unique, Required	Login handle
password_hash	String	Required	Encrypted password
email	String	Required	Contact email
role	String	Enum: ['student',...]	Access rights
createdAt	Date	Default: Date.now	Creation time

## B. Rooms Collection

Field	Type	Constraints	Description
_id	ObjectId	Auto-generated	Unique ID
roomNumber	String	Unique, Required	e.g., "101B"
capacity	Integer	Required	Max people
type	String	Enum: ['normal'...]	Filter category
features	Array	Optional	e.g., Projector

## C. BaseSchedule Collection (Official Timetable)

Field	Type	Constraints	Description
_id	ObjectId	Auto-generated	Unique ID
room_id	ObjectId	Ref: Rooms	Room Link
subject	String	Required	Class Name
dayOfWeek	String	Enum: ['Mon'...]	Repeats on
startTime	String	24h format	Start (08:30)
endTime	String	24h format	End (10:00)
semester	String	Required	Current Term

# University Room Booking - Database Configuration

## D. Bookings Collection (Dynamic)

Field	Type	Constraints	Description
_id	ObjectId	Auto-generated	Unique ID
room_id	ObjectId	Ref: Rooms	Room Link
user_id	ObjectId	Ref: Users	User Link
startTime	Date	Required	ISO Timestamp
endTime	Date	Required	ISO Timestamp
status	String	Enum: ['confirmed'...]	Booking State
overriddenBy	ObjectId	Ref: Users	Teacher Override

## 3. Priority Logic (Teacher Override)

Conflict Detection: (newStart < existingEnd) AND (newEnd > existingStart)

Booking Request Flow:

1. Check Base Schedule for official classes -> Block if conflict
2. Check Existing Bookings:
  - a. If empty -> Allow
  - b. If booked by teacher -> Reject (Teachers have priority)
  - c. If booked by student & requester is teacher -> Override/Cancel
  - d. If booked by student & requester is student -> Reject (max 2 hours)

## 4. Database Indexing & Optimization

### Room Indexes:

- { roomNumber: 1 } - Fast lookup by room code
- { type: 1, capacity: 1 } - Filter by attributes

### Booking Indexes:

- { room\_id: 1, startTime: 1, endTime: 1 } - Conflict detection
- { user\_id: 1, startTime: -1 } - User history & 2-hour check