**Doctor Management System**

**Final Report**

**Member1**

**Member2**

**Member3**

**Member4**

**Project Overview**

The Doctor Management System is a comprehensive web-based platform designed to streamline healthcare operations by connecting administrators, doctors, patients, and staff through a centralized, efficient interface. This system is intended to modernize clinic or hospital workflows, offering improved scheduling, appointment tracking, and medical record management while reducing administrative overhead. It is structured with role-specific functionalities to ensure that each user type—Admin, Doctor, Patient, and Staff—has access to only the tools they need, improving both data security and user experience.

At its core, the system empowers the **admin** to manage medical personnel effectively. Admins can add and modify doctor profiles and have the authority to assign or change the designation of a doctor to "primary doctor," likely indicating a lead or featured role within the practice. In addition to managing doctors, admins also handle staff member data, ensuring that operational roles are kept up to date. Another critical function for the admin is the ability to view patient records, allowing oversight and high-level decision-making.

**Doctors** using the system are provided with a complete overview of their appointments. They can view all reservations and add or edit medical notes and prescriptions linked to individual appointments. This capability ensures that all patient interactions are documented digitally for continuity of care. Moreover, doctors can define their availability on a weekly basis, which the system will automatically segment into 20-minute booking slots for patients. This automated breakdown not only saves time but also standardizes the appointment length, ensuring consistent patient throughput.

**Patients** benefit from a streamlined booking experience. They can view real-time doctor availability and make appointments based on their preferred timing. Once booked, patients have access to a historical record of all their appointments, along with any notes or prescriptions provided during those visits. This feature is essential for transparency, better health monitoring, and avoiding redundant consultations or prescriptions.

**Staff** members act as the bridge between patients and healthcare providers at the point of service. Their role in the system is to monitor scheduled appointments and handle the check-in process. This includes confirming patient arrivals and collecting any due payments before the consultation. Having this functionality digitized reduces the reliance on paper-based logs and minimizes errors related to billing and appointment tracking.

In summary, the Doctor Management System is a well-rounded solution aimed at simplifying clinic operations while enhancing the patient and provider experience. It introduces efficiency into administrative processes, fosters better patient engagement through transparency, and equips healthcare providers with tools that support accurate, timely care delivery. By automating essential functions and offering tailored interfaces for each user role, the system ensures that all stakeholders—from backend staff to frontline doctors—can perform their duties with greater speed, accuracy, and coordination.

**Database Description**

The Doctor Management System database is designed to manage the core operations of a healthcare facility, enabling seamless administration of doctors, staff, appointments, patients, payments, and scheduling. It utilizes MongoDB as the database engine with Prisma as the ORM to provide a scalable, structured, and efficient way of handling clinical data. The schema is optimized for multi-role usage with secure access controls and clear relationships among all medical, administrative, and financial records.

**Key Entities and Relationships**

**Admin**  
• *Represents*: System administrators  
• *Attributes*:

* id (PK): Unique identifier (ObjectId)
* email: Unique login credential
* password: Hashed password
* firstName, lastName: Personal identity
* streetAddress, city, state, zipCode: Full address
* dob: Date of birth
* phone: Contact number
* gender: Gender of the admin
* role: User role (ADMIN by default)
* createdAt, updatedAt: Timestamps for auditing

**Doctor**  
• *Represents*: Healthcare providers  
• *Attributes*:

* id (PK): Unique identifier (ObjectId)
* email, password: Secure credentials
* firstName, lastName: Name details
* ssn: Unique Social Security Number
* streetAddress, city, state, zipCode: Location details
* dob, phone, gender: Demographic information
* isPrimary: Flag for primary doctor designation
* role: Defaulted to DOCTOR
* hasResetPassword: Boolean for password reset status
* createdAt, updatedAt: Audit fields  
  • *Relationships*:
* Can have multiple appointments, availability records, and patientSlots

**Staff**  
• *Represents*: Operational staff members  
• *Attributes*:

* id (PK): Unique identifier (ObjectId)
* email, password: Login credentials
* ssn: Unique identifier
* firstName, lastName, phone, dob, gender: Personal info
* streetAddress, city, state, zipCode: Address
* role: STAFF by default
* hasResetPassword: Password status tracking
* createdAt, updatedAt: Timestamps  
  • *Relationships*:
* Can check-in multiple appointments

**Patient**  
• *Represents*: Registered patients  
• *Attributes*:

* id (PK): Unique identifier (ObjectId)
* email, password: Authentication
* firstName, lastName, phone, dob, gender: Identity and contact
* streetAddress, city, state, zipCode: Address
* height, weight: Health metrics
* role: PATIENT by default
* createdAt, updatedAt: Lifecycle tracking  
  • *Relationships*:
* Has multiple appointments and payments

**Appointment**  
• *Represents*: Booked appointments between doctors and patients  
• *Attributes*:

* id (PK): Unique identifier (ObjectId)
* fee: Consultation fee
* type: AppointmentType (e.g., CONSULTATION, TEST)
* status: AppointmentStatus (e.g., CONFIRMED, CHECKED\_IN)
* notes, prescription: Optional medical notes
* createdAt, updatedAt: Activity timestamps  
  • *Relationships*:
* Linked to one patient, one doctor, and one slot
* Optionally linked to staff (for check-in) and one payment

**DoctorAvailability**  
• *Represents*: Weekly availability schedule of a doctor  
• *Attributes*:

* id (PK): Unique identifier (ObjectId)
* day: Day of the week
* startTime, endTime: Time bounds for availability
* createdAt, updatedAt: Tracking fields  
  • *Relationships*:
* Belongs to one doctor
* Can generate multiple patientSlots

**PatientSlots**  
• *Represents*: 20-minute appointment slots generated from doctor availability  
• *Attributes*:

* id (PK): Unique identifier (ObjectId)
* day, startTime, endTime: Time allocation
* createdAt, updatedAt: Lifecycle fields  
  • *Relationships*:
* Belongs to a doctor and doctorAvailability
* Can have multiple appointments

**Payment**  
• *Represents*: Financial transactions linked to appointments  
• *Attributes*:

* id (PK): Unique identifier (ObjectId)
* paymentMethod: Enum for payment type (e.g., CREDIT\_CARD, CASH)
* amount: Payment value
* insuranceNumber: Optional insurance reference
* createdAt, updatedAt: Transaction lifecycle  
  • *Relationships*:
* Linked to one appointment (1:1)
* Optionally linked to a patient

**Core Functionality Reflected in Schema**

* **Role-Based User Management**: Clearly defined roles (ADMIN, DOCTOR, STAFF, PATIENT) with segregated access and data responsibilities.
* **Availability Scheduling and Slot Generation**: Doctors input availability which automatically creates 20-minute slots stored in PatientSlots for patient selection.
* **Appointment Lifecycle**: Appointments are rich entities connected to slots, patients, doctors, payments, and optionally staff, ensuring complete traceability.
* **Payment Integration**: Supports multiple payment methods and insurance details for flexible and auditable transaction processing.
* **Medical Record Keeping**: Notes and prescriptions are directly embedded in the appointment model for seamless health record management.
* **Secure and Auditable**: Timestamps and unique constraints ensure high data integrity and traceability across the system.

This schema underpins the functionality of the Doctor Management System and supports operational, clinical, and administrative tasks with clarity, consistency, and extensibility.

**User Roles**

**Administrator Role**  
The Administrator (Admin) role in the Doctor Management System holds the highest level of control and access across the platform. Admins are responsible for managing both personnel and patients, ensuring the system remains organized, secure, and up to date. They have the authority to create, edit, and delete doctor and staff profiles, including the ability to designate certain doctors as "Primary Doctors" for special responsibilities or visibility. Admins oversee the operational structure of the clinic or healthcare facility by maintaining accurate user records and monitoring user access.

In addition to managing healthcare providers and staff, administrators also have visibility into the patient database. They can view patient profiles and health records, though their role does not include medical note-taking or prescription management. This oversight is essential for ensuring compliance, managing internal workflows, and supporting organizational needs such as audits or high-level decision-making.

Admins also play a pivotal role in system maintenance, user account troubleshooting, and security management. They ensure that doctors and staff have the correct access privileges and that system updates or configurations are aligned with the clinic’s policies. Admins do not interact directly with appointments or billing but have the necessary insight to oversee these processes. Their comprehensive access helps maintain operational integrity, enforce protocols, and facilitate smooth coordination between all user types.

**Doctor Role**  
The Doctor role is central to the medical operations within the system. Doctors are responsible for managing their own schedule, reviewing appointments, and maintaining accurate medical records. They can input their weekly availability, which is automatically converted by the system into 20-minute booking slots for patient appointments. This automation reduces administrative burden and allows for consistent, manageable scheduling across all doctors.

Doctors have access to a complete history of their patient appointments. For each reservation, they can view patient details and are authorized to add or edit medical notes and prescriptions. These records serve as essential tools for diagnosis, treatment planning, and continuity of care. Doctors are not permitted to access administrative tools, patient billing information, or the broader staff and patient directories.

Their role is strictly clinical, ensuring a focused and secure environment in which to deliver care. The system supports doctors by providing structured tools for documentation, scheduling, and communication, enhancing their ability to deliver timely, personalized care while maintaining accurate and secure records.

**Patient Role**  
The Patient role represents the end-users of the healthcare system—those receiving medical care. Patients are able to search for doctors based on real-time availability and book appointments directly through the platform. Once an appointment is confirmed, patients can view the details of their upcoming and past consultations, including any notes or prescriptions that the doctor has added.

Patients manage their own profile, including contact information and health metrics such as height and weight. They also have access to their billing history and associated payments, particularly when linked to insurance or direct transactions made during check-in.

Patients cannot access or view other users’ information, including doctors’ or other patients’ data beyond what is explicitly shared through appointments. Their interaction is secure, streamlined, and limited to personal medical and scheduling information, promoting privacy while supporting proactive health management.

**Staff Role**  
The Staff role serves as a critical bridge between administrative and clinical workflows. Staff members are responsible for managing on-site patient interactions, primarily during the check-in phase of appointments. They are able to view scheduled appointments and confirm patient arrivals by processing payments and updating appointment statuses (e.g., marking a patient as checked-in).

While staff members do not have access to private medical data such as prescriptions or doctor's notes, they are essential to maintaining the operational flow of the clinic. Their responsibilities focus on logistical and transactional aspects of patient visits, ensuring a smooth transition from appointment booking to consultation.

Staff also assist in verifying patient information and may support system users with general inquiries. However, they are restricted from modifying user roles, editing medical records, or accessing system-wide reports, maintaining a clear separation of duties that supports both security and operational efficiency.

**Security and Access Control**  
The Doctor Management System implements strong role-based access control to ensure each user only accesses data and functionalities relevant to their responsibilities. Credentials are securely stored and protected by strict authentication protocols, including hashed passwords and session management. Admins have the broadest access rights, followed by doctors and staff, with patients having the most restricted access, focused solely on their personal data.

Every interaction is monitored and logged, particularly actions by privileged users such as admins and doctors. This approach supports regulatory compliance, maintains accountability, and ensures the integrity of all health records and transactions. The defined user roles and access limitations form the foundation of a secure, efficient, and trustworthy healthcare management environment.

**Customer Audience**

The Doctor Management System is built to serve multiple stakeholders across the healthcare delivery spectrum. Each audience segment has distinct goals, operational challenges, and expectations, which the system addresses through carefully crafted role-specific functionalities. From solo practitioners and busy front desks to empowered patients and organizational leaders, this platform is designed for efficiency, clarity, and scalable healthcare management.

**1. Healthcare Providers (Doctors, Specialists, Medical Consultants)**

Doctors are core users of the system, and the platform is designed to support their clinical and time-management needs. The system allows them to view and manage all scheduled appointments in one place, minimizing the need for fragmented tools or paper records. They can enter consultation notes, prescribe medications, and maintain detailed histories for each patient they attend to. Weekly availability can be inputted in advance, and the system automatically translates this into structured 20-minute time slots—eliminating the need for manual scheduling.

This role is especially valuable for:

* **Private practitioners** managing their own schedules
* **Group practices** with shared patient loads
* **Specialists** (e.g., dermatologists, cardiologists, orthopedists) who handle a large number of targeted consultations
* **Telehealth doctors** needing real-time access to notes, availability, and booking

**2. Clinic and Hospital Administrators**  
Administrators are the primary operational overseers. This audience includes medical office managers, hospital IT coordinators, or senior administrative personnel who are responsible for staffing, system management, compliance, and strategic oversight. The system allows them to onboard and manage doctors and staff members, designate primary doctors, and monitor all registered patients.

Key tasks for this group include:

* Maintaining an accurate and up-to-date medical personnel database
* Viewing patient demographics and usage metrics for audits and compliance
* Assigning operational roles and ensuring the right access controls
* Supporting infrastructure that enables doctors and staff to function efficiently

**3. Front Office & Support Staff (Receptionists, Check-in Desk, Billing Coordinators)**  
Support staff members play a crucial role in patient-facing processes. They use the system primarily to manage appointments and patient check-ins. This includes confirming arrivals, collecting and recording payments, and updating appointment statuses (e.g., "checked-in"). Their work ensures that doctors and patients can focus on care delivery while logistics are handled smoothly behind the scenes.

Staff users typically work in:

* **Reception desks** where they greet and register walk-in patients
* **Billing departments** where they handle cash, card, or insurance transactions
* **Call centers or help desks** responsible for verifying bookings and resolving patient queries

The system gives them a focused dashboard with limited but powerful tools—reducing the risk of over-access while enhancing productivity.

**4. Patients and Healthcare Consumers**  
Patients are the largest audience for the system and the ultimate beneficiaries of its efficiency. They can easily find doctors based on availability, book appointments in real time, and access their historical records. After each appointment, patients can view their medical notes, prescribed medications, and payment receipts. These features promote proactive health management and offer a transparent care experience.

This audience may include:

* **Busy working professionals** looking for flexible online booking
* **Parents managing appointments for children or elderly family members**
* **Chronic care patients** who need consistent follow-ups and prescription tracking
* **New patients** seeking information about available doctors and consultation types

By giving patients visibility into their healthcare journey, the system supports better decision-making, fewer missed appointments, and stronger continuity of care.

**5. Independent Clinics, Specialty Practices, and Mid-Sized Hospitals**  
Beyond individual roles, the Doctor Management System is ideal for entire **medical institutions**, especially those operating without an in-house IT team or large-scale EMR (Electronic Medical Record) system. These clinics may lack the budget or need for complex hospital-grade solutions but still require a modern, secure, and reliable system to manage operations.

Institutions benefiting from this solution include:

* **Specialty clinics** like fertility centers, ENT clinics, or mental health facilities
* **Daycare surgery centers** needing scheduling and patient coordination
* **Rural or semi-urban health centers** where automation and patient tracking are critical
* **Group practices** needing centralized booking with role-based control

These organizations often value simplicity, affordability, speed of deployment, and support for multi-role workflows—all of which this system is designed to deliver.

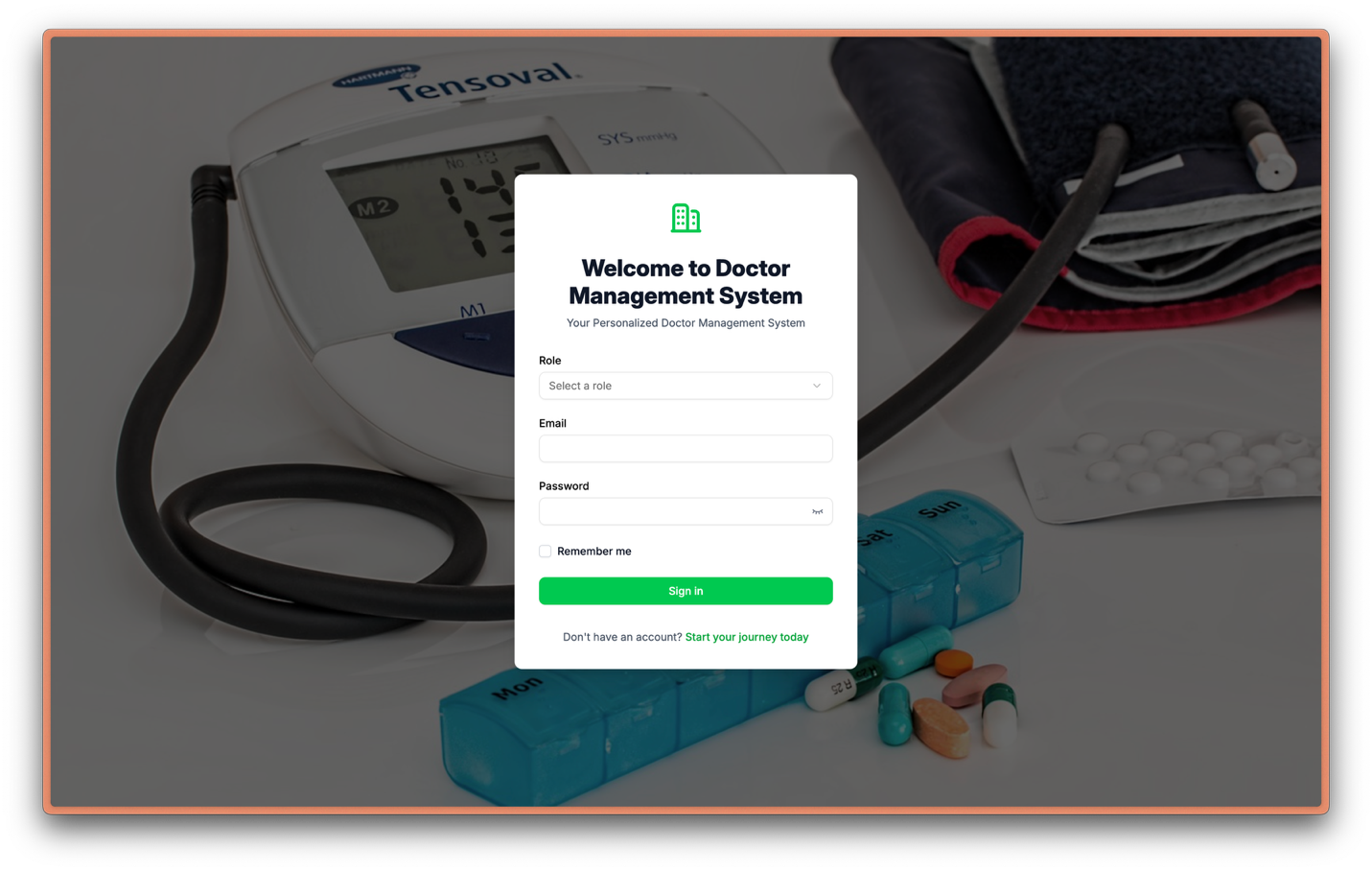
**Summary**

The Doctor Management System is a robust, role-based healthcare software solution tailored to streamline the day-to-day operations of medical clinics, hospitals, and private practices. Built with a modern tech stack using MongoDB and Prisma ORM, the system facilitates efficient interaction among administrators, doctors, staff, and patients—each with clearly defined access and functionality. Its design ensures secure data handling, seamless scheduling, transparent medical record access, and simplified billing processes.

Administrators can manage and monitor doctors, staff, and patients while maintaining overall control of the system’s structure. Doctors benefit from tools that allow them to manage their appointments, document patient notes, prescribe treatments, and define their weekly availability, which the system automatically breaks down into 20-minute booking slots. Staff members handle operational tasks like check-ins and payment collection, enabling smoother front-desk workflows. Patients enjoy a user-friendly interface to view doctor availability, book appointments, and access their health records, prescriptions, and billing history.

The system caters to a diverse audience, including solo practitioners, specialty clinics, mid-sized healthcare facilities, support personnel, and patients of all backgrounds. With a focus on role-based access control, automation, and data integrity, the Doctor Management System improves clinical productivity, enhances patient engagement, and ensures compliance with healthcare standards. It is a comprehensive, scalable solution designed to modernize and optimize healthcare service delivery across various organizational settings.

**Screenshots**

****

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a calendar

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**ER Diagram**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Sample Data**

{

"admin": {

"id": "admin1",

"firstName": "Alice",

"lastName": "Wong",

"streetAddress": "123 Admin St",

"city": "Adminville",

"state": "CA",

"zipCode": "90001",

"dob": "1980-05-15T00:00:00.000Z",

"phone": "123-456-7890",

"gender": "FEMALE",

"email": "alice@clinic.com",

"password": "securePassword123",

"role": "ADMIN",

"createdAt": "2025-01-01T10:00:00.000Z",

"updatedAt": "2025-01-01T10:00:00.000Z"

},

"doctor": {

"id": "doctor1",

"firstName": "Bob",

"lastName": "Smith",

"streetAddress": "456 Doctor Ave",

"city": "Docville",

"state": "NY",

"zipCode": "10001",

"dob": "1975-09-20T00:00:00.000Z",

"phone": "321-654-0987",

"gender": "MALE",

"ssn": "123-45-6789",

"email": "bob@clinic.com",

"password": "passwordDoc123",

"isPrimary": true,

"role": "DOCTOR",

"hasResetPassword": false,

"createdAt": "2025-01-01T10:00:00.000Z",

"updatedAt": "2025-01-01T10:00:00.000Z"

},

"staff": {

"id": "staff1",

"firstName": "Carol",

"lastName": "Jones",

"streetAddress": "789 Staff Blvd",

"city": "Staffcity",

"state": "TX",

"zipCode": "73301",

"dob": "1990-03-10T00:00:00.000Z",

"phone": "555-555-5555",

"gender": "FEMALE",

"ssn": "987-65-4321",

"email": "carol@clinic.com",

"password": "staffPass123",

"role": "STAFF",

"hasResetPassword": false,

"createdAt": "2025-01-01T10:00:00.000Z",

"updatedAt": "2025-01-01T10:00:00.000Z"

},

"patient": {

"id": "patient1",

"firstName": "David",

"lastName": "Lee",

"streetAddress": "321 Patient Rd",

"city": "Patientown",

"state": "WA",

"zipCode": "98001",

"dob": "2000-01-01T00:00:00.000Z",

"phone": "444-444-4444",

"gender": "MALE",

"height": 175.5,

"weight": 70.2,

"email": "david@patient.com",

"password": "patient123",

"role": "PATIENT",

"createdAt": "2025-01-01T10:00:00.000Z",

"updatedAt": "2025-01-01T10:00:00.000Z"

},

"doctorAvailability": {

"id": "avail1",

"day": "MONDAY",

"startTime": "2025-05-12T09:00:00.000Z",

"endTime": "2025-05-12T17:00:00.000Z",

"doctorId": "doctor1",

"createdAt": "2025-01-01T10:00:00.000Z",

"updatedAt": "2025-01-01T10:00:00.000Z"

},

"patientSlots": {

"id": "slot1",

"day": "MONDAY",

"startTime": "2025-05-12T09:00:00.000Z",

"endTime": "2025-05-12T09:30:00.000Z",

"doctorId": "doctor1",

"availabilityId": "avail1",

"createdAt": "2025-01-01T10:00:00.000Z",

"updatedAt": "2025-01-01T10:00:00.000Z"

},

"appointment": {

"id": "appoint1",

"fee": 150.0,

"type": "CONSULTATION",

"status": "CONFIRMED",

"notes": "Initial checkup",

"prescription": null,

"patientId": "patient1",

"doctorId": "doctor1",

"slotId": "slot1",

"staffId": "staff1",

"createdAt": "2025-01-01T10:00:00.000Z",

"updatedAt": "2025-01-01T10:00:00.000Z"

},

"payment": {

"id": "payment1",

"paymentMethod": "CREDIT\_CARD",

"amount": 150.0,

"insuranceNumber": null,

"appointmentId": "appoint1",

"patientId": "patient1",

"createdAt": "2025-01-01T10:00:00.000Z",

"updatedAt": "2025-01-01T10:00:00.000Z"

}

}

**MongoDB Shell Commands**

// Connect to your database first using `use your\_database\_name`

use clinicDB;

// Insert Admin

db.admin.insertOne({

\_id: ObjectId("664a4a0a0000000000000001"),

firstName: "Alice",

lastName: "Wong",

streetAddress: "123 Admin St",

city: "Adminville",

state: "CA",

zipCode: "90001",

date\_of\_birth: ISODate("1980-05-15T00:00:00Z"),

phone: "123-456-7890",

gender: "FEMALE",

email: "alice@clinic.com",

password: "securePassword123",

role: "ADMIN",

createdAt: ISODate("2025-01-01T10:00:00Z"),

updatedAt: ISODate("2025-01-01T10:00:00Z")

});

// Insert Doctor

db.doctor.insertOne({

\_id: ObjectId("664a4a0a0000000000000002"),

firstName: "Bob",

lastName: "Smith",

streetAddress: "456 Doctor Ave",

city: "Docville",

state: "NY",

zipCode: "10001",

date\_of\_birth: ISODate("1975-09-20T00:00:00Z"),

phone: "321-654-0987",

gender: "MALE",

ssn: "123-45-6789",

email: "bob@clinic.com",

password: "passwordDoc123",

isPrimary: true,

role: "DOCTOR",

hasResetPassword: false,

createdAt: ISODate("2025-01-01T10:00:00Z"),

updatedAt: ISODate("2025-01-01T10:00:00Z")

});

// Insert Staff

db.staff.insertOne({

\_id: ObjectId("664a4a0a0000000000000003"),

firstName: "Carol",

lastName: "Jones",

streetAddress: "789 Staff Blvd",

city: "Staffcity",

state: "TX",

zipCode: "73301",

dob: ISODate("1990-03-10T00:00:00Z"),

phone: "555-555-5555",

gender: "FEMALE",

ssn: "987-65-4321",

email: "carol@clinic.com",

password: "staffPass123",

role: "STAFF",

hasResetPassword: false,

createdAt: ISODate("2025-01-01T10:00:00Z"),

updatedAt: ISODate("2025-01-01T10:00:00Z")

});

// Insert Patient

db.patient.insertOne({

\_id: ObjectId("664a4a0a0000000000000004"),

firstName: "David",

lastName: "Lee",

streetAddress: "321 Patient Rd",

city: "Patientown",

state: "WA",

zipCode: "98001",

dob: ISODate("2000-01-01T00:00:00Z"),

phone: "444-444-4444",

gender: "MALE",

height: 175.5,

weight: 70.2,

email: "david@patient.com",

password: "patient123",

role: "PATIENT",

createdAt: ISODate("2025-01-01T10:00:00Z"),

updatedAt: ISODate("2025-01-01T10:00:00Z")

});

// Insert DoctorAvailability

db.doctor\_availability.insertOne({

\_id: ObjectId("664a4a0a0000000000000005"),

day: "MONDAY",

startTime: ISODate("2025-05-12T09:00:00Z"),

endTime: ISODate("2025-05-12T17:00:00Z"),

doctorId: ObjectId("664a4a0a0000000000000002"),

createdAt: ISODate("2025-01-01T10:00:00Z"),

updatedAt: ISODate("2025-01-01T10:00:00Z")

});

// Insert PatientSlots

db.patient\_slots.insertOne({

\_id: ObjectId("664a4a0a0000000000000006"),

day: "MONDAY",

startTime: ISODate("2025-05-12T09:00:00Z"),

endTime: ISODate("2025-05-12T09:30:00Z"),

doctorId: ObjectId("664a4a0a0000000000000002"),

availabilityId: ObjectId("664a4a0a0000000000000005"),

createdAt: ISODate("2025-01-01T10:00:00Z"),

updatedAt: ISODate("2025-01-01T10:00:00Z")

});

// Insert Appointment

db.appointment.insertOne({

\_id: ObjectId("664a4a0a0000000000000007"),

fee: 150.0,

type: "CONSULTATION",

status: "CONFIRMED",

notes: "Initial checkup",

prescription: null,

patientId: ObjectId("664a4a0a0000000000000004"),

doctorId: ObjectId("664a4a0a0000000000000002"),

slotId: ObjectId("664a4a0a0000000000000006"),

staffId: ObjectId("664a4a0a0000000000000003"),

createdAt: ISODate("2025-01-01T10:00:00Z"),

updatedAt: ISODate("2025-01-01T10:00:00Z")

});

// Insert Payment

db.payment.insertOne({

\_id: ObjectId("664a4a0a0000000000000008"),

paymentMethod: "CREDIT\_CARD",

amount: 150.0,

insuranceNumber: null,

appointmentId: ObjectId("664a4a0a0000000000000007"),

patientId: ObjectId("664a4a0a0000000000000004"),

createdAt: ISODate("2025-01-01T10:00:00Z"),

updatedAt: ISODate("2025-01-01T10:00:00Z")

});

// Read Data

print("Admins:");

db.admin.find().pretty();

print("Doctors:");

db.doctor.find().pretty();

print("Staff:");

db.staff.find().pretty();

print("Patients:");

db.patient.find().pretty();

print("Doctor Availabilities:");

db.doctor\_availability.find().pretty();

print("Patient Slots:");

db.patient\_slots.find().pretty();

print("Appointments:");

db.appointment.find().pretty();

print("Payments:");

db.payment.find().pretty();