

## Database Schema for Consultation Database

**address**(id VARCHAR(255), address\_line1 VARCHAR(255), address\_line2 VARCHAR(255), city VARCHAR(255), postcode VARCHAR(255), state VARCHAR(255))

**appointment**(appointment\_id VARCHAR(255), appointment\_date VARCHAR(255), created\_date VARCHAR(255), doctor\_id VARCHAR(255), doctor\_name VARCHAR(255), prior\_medical\_history VARCHAR(255), status VARCHAR(255), symptoms VARCHAR(255), time\_slot VARCHAR(255), user\_email\_id VARCHAR(255), user\_id VARCHAR(255), user\_name VARCHAR(255))

**doctor**(id VARCHAR(255), college VARCHAR(255), dob VARCHAR(255), email\_id VARCHAR(255), first\_name VARCHAR(255), highest\_qualification VARCHAR(255), last\_name VARCHAR(255), mobile VARCHAR(255), pan VARCHAR(255), rating DOUBLE, speciality VARCHAR(255), total\_years\_of\_exp INT)

**rating**(id VARCHAR(255), appointment\_id VARCHAR(255), comments VARCHAR(255), doctor\_id VARCHAR(255), rating INT)

**user**(email\_id VARCHAR(255), created\_date VARCHAR(255), dob VARCHAR(255), first\_name VARCHAR(255), last\_name VARCHAR(255), mobile VARCHAR(255), password VARCHAR(255), salt VARCHAR(255))

**user\_auth\_token**(id BIGINT, access\_token VARCHAR(1000), expires\_at DATETIME, login\_at DATETIME, logout\_at DATETIME, user\_id VARCHAR(255))

appointment	
🔑	appointment_id VARCHAR(255)
🔑	appointment_date VARCHAR(255)
🔑	created_date VARCHAR(255)
🔑	doctor_id VARCHAR(255)
🔑	doctor_name VARCHAR(255)
🔑	prior_medical_history VARCHAR(255)
🔑	status VARCHAR(255)
🔑	symptoms VARCHAR(255)
🔑	time_slot VARCHAR(255)
🔑	user_email_id VARCHAR(255)
🔑	user_id VARCHAR(255)
🔑	user_name VARCHAR(255)
Indexes ▶	

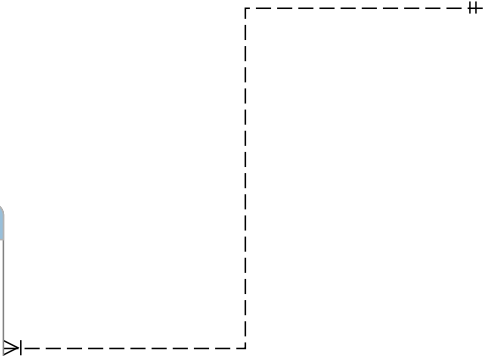
address	
🔑	id VARCHAR(255)
🔑	address_line1 VARCHAR(255)
🔑	address_line2 VARCHAR(255)
🔑	city VARCHAR(255)
🔑	postcode VARCHAR(255)
🔑	state VARCHAR(255)
Indexes ▶	

doctor	
🔑	id VARCHAR(255)
🔑	college VARCHAR(255)
🔑	dob VARCHAR(255)
🔑	email_id VARCHAR(255)
🔑	first_name VARCHAR(255)
🔑	highest_qualification VARCHAR(255)
🔑	last_name VARCHAR(255)
🔑	mobile VARCHAR(255)
🔑	pan VARCHAR(255)
🔑	rating DOUBLE
🔑	speciality VARCHAR(255)
🔑	total_years_of_exp INT
Indexes ▶	

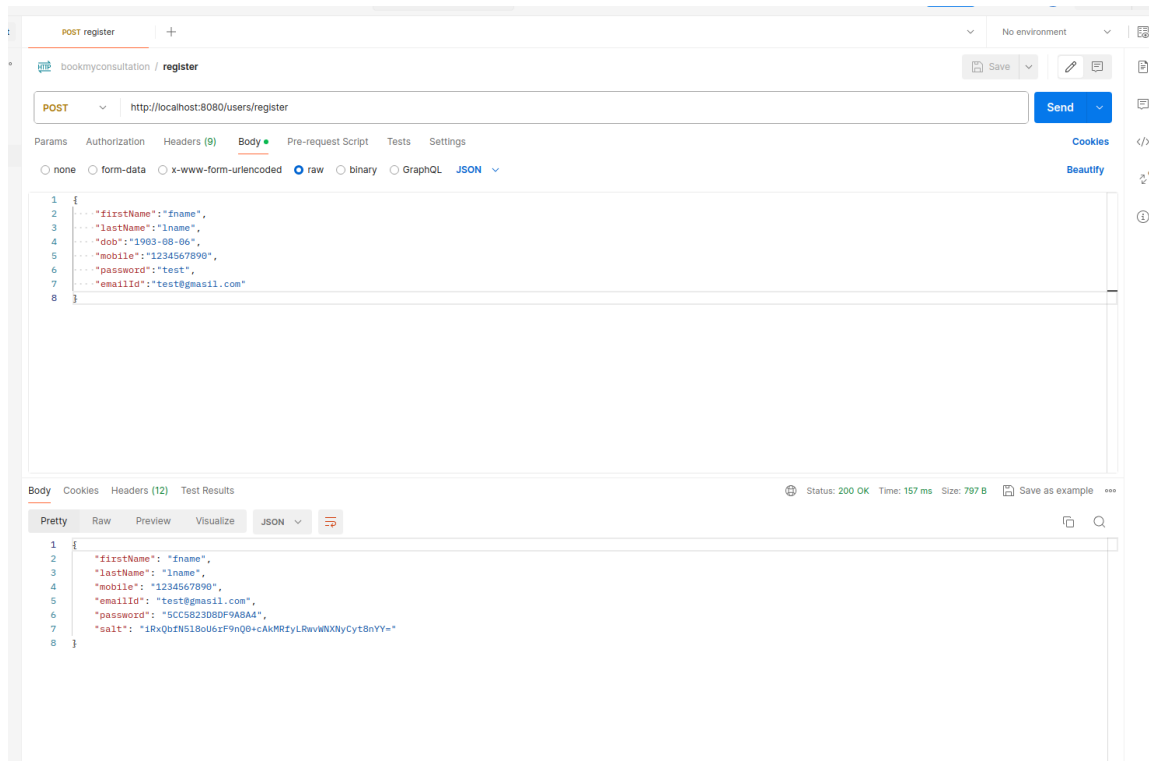
user	
🔑	email_id VARCHAR(255)
🔑	created_date VARCHAR(255)
🔑	dob VARCHAR(255)
🔑	first_name VARCHAR(255)
🔑	last_name VARCHAR(255)
🔑	mobile VARCHAR(255)
🔑	password VARCHAR(255)
🔑	salt VARCHAR(255)
Indexes ▶	

user_auth_token	
🔑	id BIGINT
🔑	access_token VARCHAR(1000)
🔑	expires_at DATETIME
🔑	login_at DATETIME
🔑	logout_at DATETIME
🔑	user_id VARCHAR(255)
Indexes ▶	

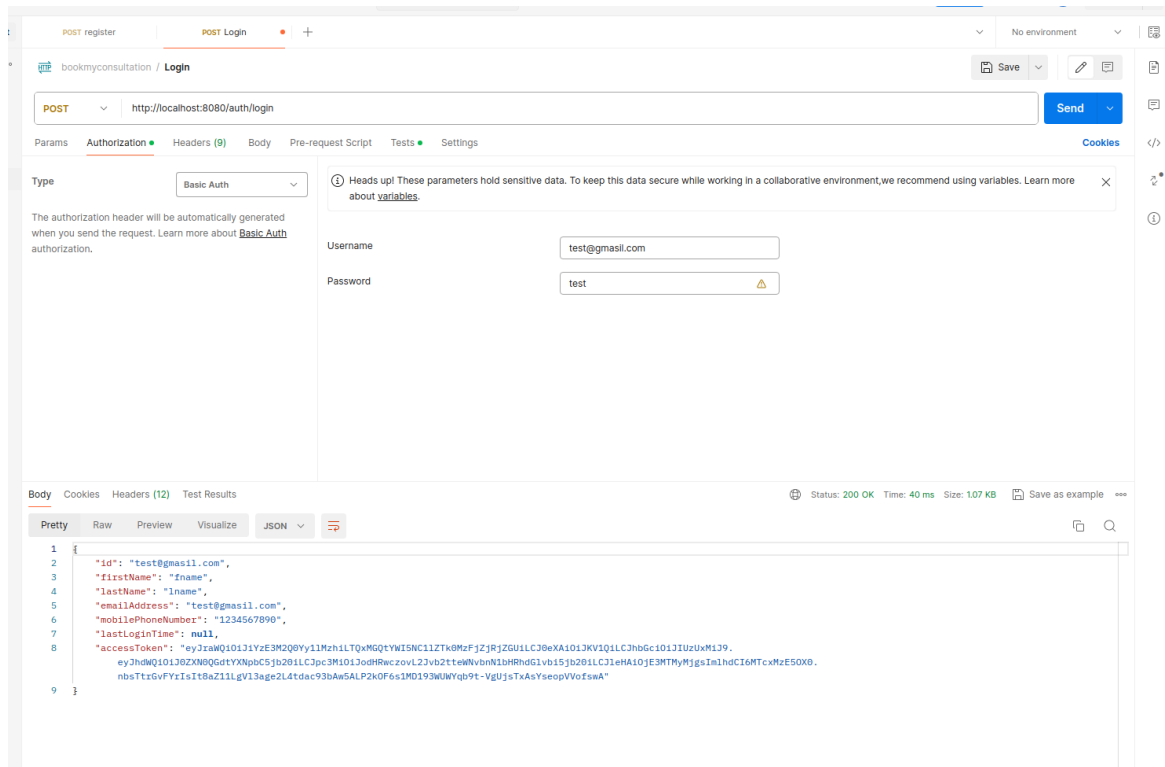
rating	
🔑	id VARCHAR(255)
🔑	appointment_id VARCHAR(255)
🔑	comments VARCHAR(255)
🔑	doctor_id VARCHAR(255)
🔑	rating INT
Indexes ▶	



API 1) POST call to register a user in the database.



API 2) POST call to login with credentials and get the Auth Token.



### API 3) GET call to fetch all Doctors

Postman interface showing a GET request to `http://localhost:8080/doctors`. The request headers include `Cache-Control`, `Postman-Token`, `Host`, `User-Agent`, `Accept`, `Accept-Encoding`, and `Connection`. The response status is 200 OK, and the body is a JSON array of two doctor objects.

```
1 {
2   {
3     "id": "UUID-94",
4     "firstName": "Ira",
5     "lastName": "Tyson",
6     "speciality": "GENERAL_PHYSICIAN",
7     "dob": "2021-01-15",
8     "address": {
9       "city": "Pabianice"
10    },
11    "mobile": "16830515 0222",
12    "emailId": "Phaselus.at@euplacerat.edu",
13    "pan": "0X294P2K9LY",
14    "totalYearsOfExp": 49,
15    "rating": 5.0
16  },
17  {
18    "id": "UUID-48",
19    "firstName": "Holmes",
20    "lastName": "Hansen",
```

### API 4) GET call to fetch Doctors by Speciality

Postman interface showing a GET request to `http://localhost:8080/doctors?speciality=CARDIOLOGIST`. The request headers include `Cache-Control`, `Postman-Token`, `Host`, `User-Agent`, `Accept`, `Accept-Encoding`, and `Connection`. The response status is 200 OK, and the body is a JSON array of two doctor objects with the speciality `CARDIOLOGIST`.

```
1 {
2   {
3     "id": "UUID-91",
4     "firstName": "Zeph",
5     "lastName": "Kelley",
6     "speciality": "CARDIOLOGIST",
7     "dob": "2020-12-28",
8     "address": {
9       "city": "Vienna"
10    },
11    "mobile": "16501201 1044",
12    "emailId": "cursus.non@faucibus.org",
13    "pan": "XGR58GZL2II",
14    "totalYearsOfExp": 22,
15    "rating": 5.0
16  },
17  {
18    "id": "UUID-83",
19    "firstName": "Ahmed",
20    "lastName": "Guy",
```

## API 5) GET call to fetch all Speciality

The screenshot shows a Postman interface for a GET request to `http://localhost:8080/doctors/speciality`. The request is saved and has a status of 200 OK. The response is displayed in the Body tab, showing a JSON array of specialities.

**Request:**

- Method: GET
- URL: `http://localhost:8080/doctors/speciality`

**Headers (8):**

Key	Value	Description
Cache-Control	no-cache	
Postman-Token	<calculated when request is sent>	
Host	<calculated when request is sent>	
User-Agent	PostmanRuntime/7.37.3	
Accept	*/*	
Accept-Encoding	gzip, deflate, br	
Connection	keep-alive	
Authorization	Bearer ((BMCAuthToken))	

**Body:**

```
1 {
2   "CARDIOLOGIST",
3   "GENERAL_PHYSICIAN",
4   "DENTIST",
5   "PULMONOLOGIST",
6   "EMT",
7   "GASTRO"
8 }
```

## API 6) GET call to fetch all Time Slots for a given user for a specific date.

The screenshot shows a Postman interface for a GET request to `http://localhost:8080/doctors/UUID-34/timeslots?date=2029-05-26`. The request is saved and has a status of 200 OK. The response is displayed in the Body tab, showing a JSON object with a list of time slots.

**Request:**

- Method: GET
- URL: `http://localhost:8080/doctors/UUID-34/timeslots?date=2029-05-26`

**Headers (8):**

Key	Value	Description
Cache-Control	no-cache	
Postman-Token	<calculated when request is sent>	
Host	<calculated when request is sent>	
User-Agent	PostmanRuntime/7.37.3	
Accept	*/*	
Accept-Encoding	gzip, deflate, br	
Connection	keep-alive	
Authorization	Bearer ((BMCAuthToken))	

**Body:**

```
1 {
2   "doctorId": "UUID-34",
3   "availableDate": "2029-05-26",
4   "timeSlot": [
5     "11AM-12AM",
6     "07PM-08PM",
7     "06PM-07PM",
8     "05PM-06PM",
9     "08PM-09PM",
10    "12AM-01PM"
11  ]
12 }
```

## API 7) GET call to fetch a Specific Doctor Details for a given doctor id

The screenshot shows a Postman interface for a GET request to `http://localhost:8080/doctors/UUID-2`. The request is saved and has a status of 200 OK. The response body is a JSON object representing a doctor's details.

**Headers (8):**

Key	Value	Description
Cache-Control	no-cache	
Postman-Token	<calculated when request is sent>	
Host	<calculated when request is sent>	
User-Agent	PostmanRuntime/7.37.3	
Accept	/*/*	
Accept-Encoding	gzip, deflate, br	
Connection	keep-alive	
Authorization	Bearer {{BMCAuthToken}}	

**Body:**

```
1 {
2   "id": "UUID-2",
3   "firstName": "Blossom",
4   "lastName": "Valentine",
5   "speciality": "PULMONOLOGIST",
6   "dob": "2001-09-16",
7   "address": {
8     "city": "Petalillo"
9   },
10  "mobile": "16030720 8091",
11  "emailId": "malesuada@lacusMaurisnon.com",
12  "pan": "KAR905JREKQ",
13  "totalYearsOfExp": 20,
14  "rating": 5.0
15 }
```

## API 8) GET call to fetch specific User Details (Authorized)

The screenshot shows a Postman interface for a GET request to `http://localhost:8080/users/test@gmail.com`. The request is saved and has a status of 200 OK. The response body is a JSON object representing a user's details.

**Headers (8):**

Key	Value	Description
Cache-Control	no-cache	
Postman-Token	<calculated when request is sent>	
Host	<calculated when request is sent>	
User-Agent	PostmanRuntime/7.37.3	
Accept	/*/*	
Accept-Encoding	gzip, deflate, br	
Connection	keep-alive	
Authorization	Bearer eyJraWQlUjYzE3M2Q0Yy1lMzhITQxMGQYVW5lNC1lZTk0MzFjZjZl...	

**Body:**

```
1 {
2   "firstName": "fname",
3   "lastName": "lname",
4   "mobile": "1234567890",
5   "emailId": "test@gmail.com",
6   "password": "5CC5823D8DF9A8A4",
7   "salt": "1RxQbZNS1BoUzF9nQ0-cAkWRfzLRevWkXNcyT8nYy="
8 }
```

## API 9) POST call to create a Doctor record (Authorized)

The screenshot shows a REST client interface with the following details:

- URL:** `http://localhost:8080/doctors`
- Method:** `POST`
- Body (raw):**

```
1 {
2   ... "firstName": "fname",
3   ... "lastName": "lname",
4   ... "dob": "1903-08-06",
5   ... "mobile": "1234567890",
6   ... "password": "test",
7   ... "emailId": "test@gmail.com",
8   ... "pan": "EDLSS5080L",
9   ... "address": {
10    ... "addressLine1": "add1",
11    ... "addressLine2": "add1",
12    ... "city": "BHN",
13    ... "state": "TN",
14    ... "postCode": "123456"
15  }
16 }
```
- Response (JSON):**

```
1 {
2   "id": "1cc28f46-7d8d-40b3-9588-c92ed6210994",
3   "firstName": "fname",
4   "lastName": "lname",
5   "speciality": "GENERAL_PHYSICIAN",
6   "dob": "1903-08-06",
7   "address": {
8     "city": "BHN"
9   },
10  "mobile": "1234567890",
11  "emailId": "test@gmail.com",
12  "pan": "EDLSS5080L"
13 }
```
- Status:** 200 OK, Time: 22 ms, Size: 853 B

## API 10) POST call to Book an Appointment (Authorized)

The screenshot shows a REST client interface with the following details:

- URL:** `http://localhost:8080/appointments`
- Method:** `POST`
- Body (raw):**

```
1 {
2   ... "doctorId": "UUID-34",
3   ... "doctorName": "Meghan Jordan",
4   ... "userId": "test@gmail.com",
5   ... "userName": "Zname",
6   ... "userEmailId": "test@gmail.com",
7   ... "timeSlot": "9PM-6PM",
8   ... "appointmentDate": "2021-05-26",
9   ... "createdDate": "",
10  ... "symptoms": "TEST",
11  ... "priorMedicalHistory": "NA"
12 }
13 {
14   ... "doctorId": "UUID-34",
15   ... "availableDate": "2021-05-26",
16 }
```
- Response (Text):**

```
1 905480c9-8fea-41ef-b46e-68dc79048c30
```
- Status:** 200 OK, Time: 21 ms, Size: 659 B

## API 11) POST call to Rate a Appointment (Authorized)

The screenshot shows the Postman interface for a POST request. The request is named "post a rating" and is sent to the URL "http://localhost:8080/ratings". The request body is a JSON object with the following fields: "appointmentId", "doctorId", "rating", and "comments". The status bar at the bottom indicates a successful response with a status of 200 OK, a time of 29 ms, and a size of 582 B.

POST http://localhost:8080/ratings

Body

```
1 {
2   "appointmentId": "90540bc9-8fea-41ef-b46e-60dc79048c30",
3   "doctorId": "UUID-34",
4   "rating": "1",
5   "comments": "asdad"
6 }
```

Status: 200 OK Time: 29 ms Size: 582 B

## API 12) GET call to fetch an Appointment Details (Authorized)

The screenshot shows the Postman interface for a GET request. The request is named "get an appointment" and is sent to the URL "http://localhost:8080/appointments/90540bc9-8fea-41ef-b46e-60dc79048c30". The request headers are displayed, including Cache-Control, Postman-Token, Host, User-Agent, Accept, Accept-Encoding, Connection, and Authorization. The response body is a JSON object with the following fields: "appointmentId", "doctorId", "doctorName", "userId", "timeSlot", "appointmentDate", "syptoms", and "priorMedicalHistory". The status bar at the bottom indicates a successful response with a status of 200 OK, a time of 7 ms, and a size of 855 B.

GET http://localhost:8080/appointments/90540bc9-8fea-41ef-b46e-60dc79048c30

Headers (8)

Key	Value	Description
Cache-Control	no-cache	
Postman-Token	<calculated when request is sent>	
Host	<calculated when request is sent>	
User-Agent	PostmanRuntime/7.37.3	
Accept	/*	
Accept-Encoding	gzip, deflate, br	
Connection	keep-alive	
Authorization	Bearer eyJraWQlUjZlM200Yy1lMzh1LTQxMQQYWSNC1ZTk0MzFjZj...	

Body

```
1 {
2   "appointmentId": "90540bc9-8fea-41ef-b46e-60dc79048c30",
3   "doctorId": "UUID-34",
4   "doctorName": "Nephan Jordan",
5   "userId": "test@gmail.com",
6   "timeSlot": "08PM-06PM",
7   "appointmentDate": "2021-05-26",
8   "syptoms": "TEST",
9   "priorMedicalHistory": "NA"
10 }
```

Status: 200 OK Time: 7 ms Size: 855 B



## API 13) GET call to fetch all User Appointments (Authorized)

bookmyconsultation / get User appointments

GET http://localhost:8080/users/test@gmail.com/appointments

Headers (8)

Key	Value	Description
Cache-Control	no-cache	
Postman-Token	<calculated when request is sent>	
Host	<calculated when request is sent>	
User-Agent	PostmanRuntime/7.37.3	
Accept	/*/*	
Accept-Encoding	gzip, deflate, br	
Connection	keep-alive	
Authorization	Bearer eyJraWQlQjYzE3M2Q0Yy1lMzh1LTQxMGQlYW5lNC1lZTk0MzFjZj...	

Body

Status: 200 OK Time: 9 ms Size: 857 B

```
1 {
2   {
3     "appointmentId": "90548bc9-8fea-41ef-bd6e-60dc79048c30",
4     "doctorId": "UID-34",
5     "doctorName": "Meghan Jordan",
6     "userId": "test@gmail.com",
7     "timeSlot": "05PM-06PM",
8     "appointmentDate": "2021-05-26",
9     "symptoms": "TEST",
10    "priorMedicalHistory": "NA"
11  }
12 }
```

## API 14) POST call to Logout a User (Authorized)

bookmyconsultation / Logout

POST http://localhost:8080/auth/logout

Headers (9)

Key	Value	Description
Cache-Control	no-cache	
Postman-Token	<calculated when request is sent>	
Content-Length	0	
Host	<calculated when request is sent>	
User-Agent	PostmanRuntime/7.37.3	
Accept	/*/*	
Accept-Encoding	gzip, deflate, br	
Connection	keep-alive	
Authorization	Bearer eyJraWQlQjYzE3M2Q0Yy1lMzh1LTQxMGQlYW5lNC1lZTk0MzFjZj...	

Body

Status: 200 OK Time: 14 ms Size: 582 B

```
1
```