User Manual – Mini Venue Booking System (Node.js + MongoDB)

 $lue{V}$ Server Running: server started ightarrow Ready to Use

1 Register (Admin/User)

- POST http://localhost:5000/api/auth/register
- Body Example:

```
For Admin:
```

```
{ "name": "Admin User", "email": "admin@example.com", "password": "123456", "role": "admin" }
```

For User:

```
{ "name": "Normal User", "email": "user@example.com", "password": "123456", "role": "user" }
```

2 Login (Get Token)

- POST http://localhost:5000/api/auth/login
- Body Example:

```
{ "email": "admin@example.com", "password": "123456" }
```

• **V** Copy the token from response → Use in Authorization: Bearer Token header.

3 Admin Only: Add Venue

- **POST** http://localhost:5000/api/venues
- Header: Bearer <admin_token>
- Body:

```
{ "name": "Hall A", "location": "Mumbai", "capacity": 100 }
```

4 Admin Only: Block Date for Venue

- POST http://localhost:5000/api/venues/<venue_id>/block
- Header: Bearer <admin token>
- Body:

```
{ "date": "2025-07-20" }
```

5 Get All Venues (Admin/User)

- **GET** http://localhost:5000/api/venues
- Header: Bearer <admin_or_user_token>

6 User Only: Book a Venue

```
• POST http://localhost:5000/api/book
```

• Header: Bearer < user_token>

• Body:

```
{ "venueId": "<venue_id>", "bookingDate": "2025-07-21" }
```

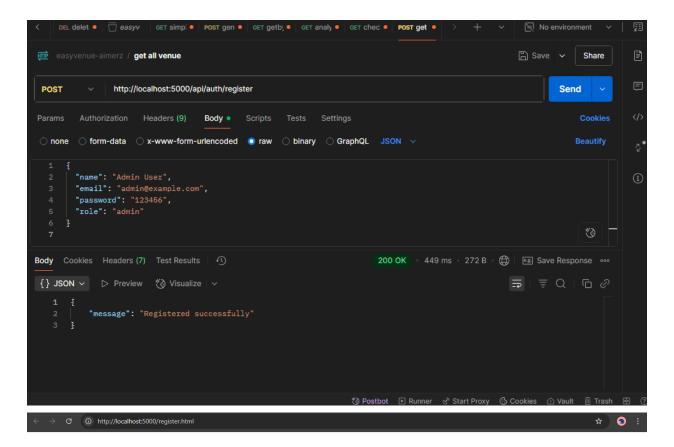
7 Frontend Testing Pages:

• **Register:** http://localhost:5000/register.html

• Login: http://localhost:5000/login.html

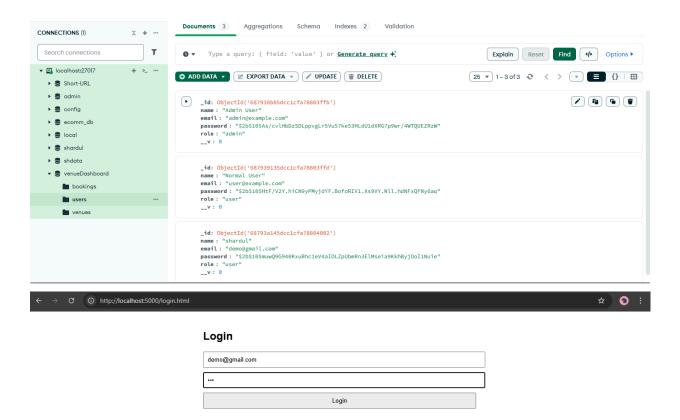
• After Login: admin.html or user.html (based on role)

Some demo screenshort



Register

| shardul | |
|----------------|----------|
| demo@gmail.com | |
| | |
| User | • |
| | Register |



User - Book Venue

ID: 687939dc5dcc1cfa78804000 | Name: Hall A

