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

✓ Server Running: **server started** → 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" }
```

-  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 screenshot

easyvenue-aimerz / **get all venue**

POST `http://localhost:5000/api/auth/register` **Send**

Params Authorization Headers (9) **Body** Scripts Tests Settings Cookies Beautify

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON**

```
1 {
2   "name": "Admin User",
3   "email": "admin@example.com",
4   "password": "123456",
5   "role": "admin"
6 }
7
```

Body Cookies Headers (7) Test Results **200 OK** · 449 ms · 272 B Save Response

{ } JSON Preview Visualize

```
1 {
2   "message": "Registered successfully"
3 }
```

Postbot Runner Start Proxy Cookies Vault Trash

`http://localhost:5000/register.html`

Register

shardul
demo@gmail.com
...
User
Register

CONNECTIONS (1)

Search connections

localhost:27017

Short-URL

admin

config

ecomm_db

local

shardul

shdata

venueDashboard

bookings

users

venues

Documents 3

Aggregations

Schema

Indexes 2

Validation

Type a query: { ffield: 'value' } or [Generate query](#)

ADD DATA

EXPORT DATA

UPDATE

DELETE

25 1 - 3 of 3

_id: ObjectId('687938b85dccc1cfa78803ffb')

name: "Admin User"

email: "admin@example.com"

password: "\$2b\$10\$As/cvLhbDzSOLppvgLr5Vu57ke53HLdU1dXRG7p9wr/4WTQEZRzW"

role: "admin"

__v: 0

_id: ObjectId('687939135dccc1cfa78803ffd')

name: "Normal User"

email: "user@example.com"

password: "\$2b\$10\$HTF/V2Y.hiCN9yPMYjdYF.BofRIV1.Xs9VY.NlL.hdNFxQFNy6aq"

role: "user"

__v: 0

_id: ObjectId('68793a145dccc1cfa78804002')

name: "shardul"

email: "demo@gmail.com"

password: "\$2b\$10\$muwQ9G940KxuBhc1eV4aIOLZpUbmRn3ELMse1a9KkhByjOoI1Nuie"

role: "user"

__v: 0

←

→

↺

🔍

http://localhost:5000/login.html

☆

🔔

⋮

Login

Login

User - Book Venue

ID: 687939dc5dcc1cfa78804000 | Name: Hall A

mm/dd/yyyy

July 2025

Su

Mo

Tu

We

Th

Fr

Sa

29

30

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

1

2

3

4

5

6

7

8

9

Clear

Today

Book Venue

Logout

Compass

My Queries

CONNECTIONS (1)

Search connections

- localhost:27017
 - Short-URL
 - admin
 - config
 - ecomm_db
 - local
 - shardul
 - shdata
 - venueDashboard
 - bookings
 - users
 - venues**

users venues +

localhost:27017 > venueDashboard > venues

Documents 1 Aggregations Schema Indexes 1 Validation

Type a query: { field: 'value' } or [Generate query](#)

ADD DATA EXPORT DATA UPDATE DELETE

```
{
  "_id": ObjectId('687939dc5dcc1cfa78804000'),
  "name": "Hall A",
  "location": "Mumbai",
  "capacity": 100,
  "blockedDates": Array (empty),
  "__v": 0
}
```

Personal Workspace New Import

easyvenue-almorz admin-getallvenue

GET http://localhost:5000/api/venues

Params Authorization Headers (8) Body Scripts Tests Settings Cookies

Auth Type: Bearer Token

The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.

Token: [REDACTED]

Body: Cookies Headers (7) Test Results 200 OK · 11 ms · 349 B Save Response

```
{
  "_id": "687939dc5dcc1cfa78804000",
  "name": "Hall A",
  "location": "Mumbai",
  "capacity": 100,
  "blockedDates": [],
  "__v": 0
}
```