It supports **CRUD** operations and is built using **Node.js**, **Express**, and **MongoDB**.

### 🟢 1. Create a New User

**Endpoint**: POST /api/users

**Description**: Adds a new user to the database.

**Request Headers**:  
Content-Type: application/json

**Request Body**:

json

CopyEdit

{

"name": "Abhay Rathore",

"email": "abhay@example.com",

"age": 25}

**Sample Response**:

json

CopyEdit

{

"\_id": "60a8cd11f0b5c00015f3e9a1",

"name": "Abhay Rathore",

"email": "abhay@example.com",

"age": 25,

"\_\_v": 0}

### 🔵 2. Get All Users

**Endpoint**: GET /api/users

**Description**: Returns a list of all users in the database.

**Request Headers**: None

**Sample Response**:

json

CopyEdit

[

{

"\_id": "60a8cd11f0b5c00015f3e9a1",

"name": "Abhay Rathore",

"email": "abhay@example.com",

"age": 25

},

{

"\_id": "60a8cd15f0b5c00015f3e9a2",

"name": "John Doe",

"email": "john@example.com",

"age": 30

}]

### 🟡 3. Update a User

**Endpoint**: PUT /api/users/:id

**Description**: Updates user information by ID.

**Request Headers**:  
Content-Type: application/json

**Request Params**:  
:id – The unique MongoDB \_id of the user to update.

**Request Body** (example to update name):

json

CopyEdit

{

"name": "Abhay Kumar"}

**Sample Response**:

json

CopyEdit

{

"\_id": "60a8cd11f0b5c00015f3e9a1",

"name": "Abhay Kumar",

"email": "abhay@example.com",

"age": 25}

### 🔴 4. Delete a User

**Endpoint**: DELETE /api/users/:id

**Description**: Deletes a user by ID.

**Request Params**:  
:id – The unique MongoDB \_id of the user to delete.

**Sample Response**:

json

CopyEdit

{

"message": "User deleted"}