

EXPERIMENT NO. - 04

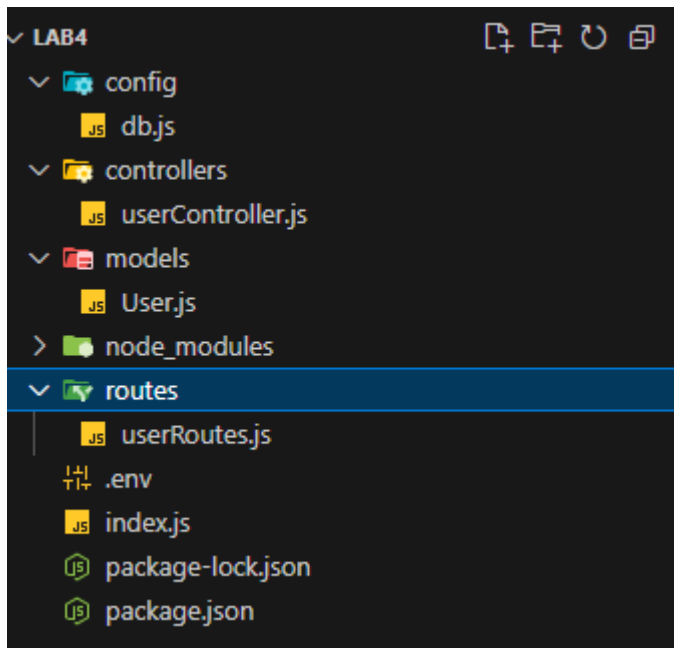
Roll No.	24
Name	Chaitanya Dinesh Dhayarkar
Class	D15B
Subject	Full Stack Development
Lab Outcome	REST API Design with MongoDB + Mongoose Integration
Date of Performance/ Submission	
Signature & Grades	

Experiment No. 4

Aim: REST API Design with MongoDB + Mongoose Integration

Theory:

Directory Structure



Code:

Index.js

```
require("dotenv").config();
const express = require("express");
const cors = require("cors");
const connectDB = require("./config/db");
const userRoutes = require("./routes/userRoutes");
const app = express();

// Middleware
app.use(cors());
app.use(express.json());
// Connect DB
connectDB();

// Routes
app.use("/users", userRoutes);

const PORT = process.env.PORT || 5000;
app.listen(PORT, () => console.log(`🚀 Server running at http://localhost:${PORT}`));
```

routes/userRoutes.js

```
routes > userRoutes.js > ...
1  const express = require("express");
2  const {
3    createUser,
4    getUsers,
5    getUserById,
6    updateUser,
7    deleteUser,
8  } = require("../controllers/userController");
9
10 const router = express.Router();
11
12 router.post("/", createUser);
13 router.get("/", getUsers);
14 router.get("/:id", getUserById);
15 router.put("/:id", updateUser);
16 router.delete("/:id", deleteUser);
17
18 module.exports = router;
19
```

Models/User.js

```
models > User.js > ...
1  const mongoose = require("mongoose");
2
3  const userSchema = new mongoose.Schema(
4    {
5      name: { type: String, required: true },
6      email: { type: String, required: true, unique: true },
7      age: { type: Number, required: true },
8    },
9    { timestamps: true }
10 );
11
12 module.exports = mongoose.model("User", userSchema);
13
```

Controllers/userController.js

```
controllers > userController.js > ...
1  const User = require("../models/User");
2
3  exports.createUser = async (req, res) => {
4    try {
5      const user = await User.create(req.body);
6      res.status(201).json({ success: true, data: user });
7    } catch (err) {
8      res.status(400).json({ success: false, message: err.message });
9    }
10   };
11
12   exports.getUsers = async (req, res) => {
13     try {
14       const users = await User.find();
15       res.json({ success: true, data: users });
16     } catch (err) {
17       res.status(500).json({ success: false, message: err.message });
18     }
19   };
20
21   exports.getUserById = async (req, res) => {
22     try {
23       const user = await User.findById(req.params.id);
24       if (!user) return res.status(404).json({ success: false, message: "User not found" });
25       res.json({ success: true, data: user });
26     } catch (err) {
27       res.status(500).json({ success: false, message: err.message });
28     }
29   };
30
31   exports.updateUser = async (req, res) => {
32     try {
33       const user = await User.findByIdAndUpdate(req.params.id, req.body, {
34         new: true,
35         runValidators: true,
36       });
37       if (!user) return res.status(404).json({ success: false, message: "User not found" });
38       res.json({ success: true, data: user });
39     }
40   }
41 }
```

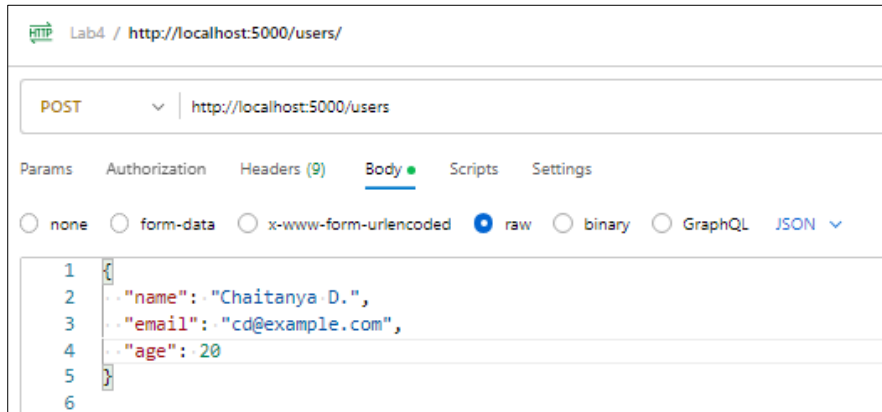
Config/db.js

```
config > db.js > ...
1  const mongoose = require("mongoose");
2
3  const connectDB = async () => {
4    console.log(process.env.MONGO_URI);
5    try {
6      await mongoose.connect(process.env.MONGO_URI);
7      console.log("✅ MongoDB Connected");
8    } catch (error) {
9      console.error("❌ MongoDB Connection Failed:", error.message);
10     process.exit(1);
11   }
12 };
13
14 module.exports = connectDB;
```

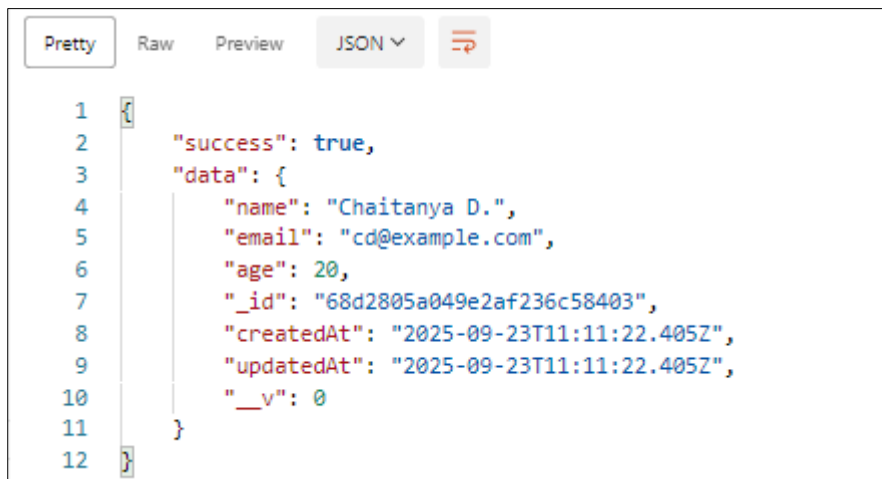
Output:

Create User (POST)

Request:

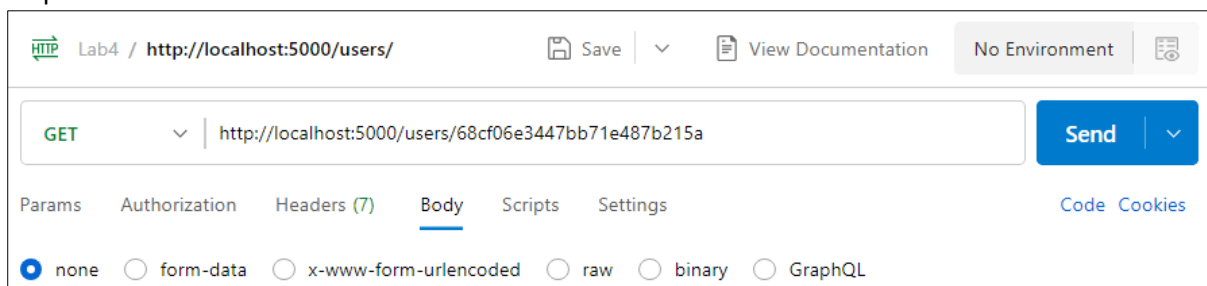


Response:



Fetch User Data (GET user by ID)

Request:



Response:

```
1  {
2    "success": true,
3    "data": {
4      "_id": "68cf06e3447bb71e487b215a",
5      "name": "Jyoti Das",
6      "email": "jd@mail.com",
7      "hashedPass": "$2b$10$Wso35RS1B2zwwEGIODPNcOR/
8      1w4e1JwxldAmcuFS90RZmEDSwxlo6",
9      "role": "user",
10     "__v": 0,
11     "lastLogin": "2025-09-20T19:56:26.499Z"
12   }
13 }
```

Fetch All Users (GET)

Request:

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

- URL: `http://localhost:5000/users/`
- Method: `GET`
- Buttons: `Save`, `View Documentation`, `No Environment`, `Send`
- Tabs: `Params`, `Authorization`, `Headers (7)`, `Body` (selected), `Scripts`, `Settings`, `Code`, `Cookies`
- Bottom Bar: `Body` (selected), `Cookies`, `Headers (8)`, `Test Results`, `Status: 200 OK`, `Time: 25 ms`, `Size: 1.32 KB`, `...`, `...`

Response:

```
1  {
2    "success": true,
3    "data": [
4      {
5        "_id": "68cf031f447bb71e487b2132",
6        "name": "Chaitanya Dinesh Dhayarkar",
7        "email": "chaitanyadhayarkar005@gmail.com",
8        "hashedPass": "$2b$10$47Hn.UMppJkxNeeu9Mdis.BKwOcTGFzpEU4YFdx9PhqvAw481Pwr.",
9        "role": "admin",
10       "__v": 0,
11       "lastLogin": "2025-09-20T20:14:24.359Z"
12     },
13     {
14       "_id": "68cf03ed447bb71e487b2148",
15       "name": "Prajwal Pawar",
16       "email": "pp@mail.com",
17       "hashedPass": "$2b$10$pEXikCxj9URAdDtR5GVguz5gjXwKbTXGXQRIPmmLY9CLvulvtX7q",
18       "role": "user",
19       "__v": 0,
20       "lastLogin": "2025-09-20T19:43:49.110Z"
21     },
22     {
23       "_id": "68cf06e3447bb71e487b215a",
```

Update User (PUT)

Request:

HTTP Lab4 / http://localhost:5000/users/

PUT http://localhost:5000/users/68d2805a049e2af236c58403

Params Authorization Headers (9) Body Scripts Settings

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

```
1 {
2   "name": "Ram D.",
3   "email": "cd@example.com",
4   "age": 18
5 }
6
```

Response:

Body Cookies Headers (8) Test Results

Pretty Raw Preview JSON

```
1 {
2   "success": true,
3   "data": {
4     "_id": "68d2805a049e2af236c58403",
5     "name": "Ram D.",
6     "email": "cd@example.com",
7     "age": 18,
8     "createdAt": "2025-09-23T11:11:22.405Z",
9     "updatedAt": "2025-09-23T11:15:18.239Z",
10    "__v": 0
11  }
12 }
```

Delete User (Delete)

Request:

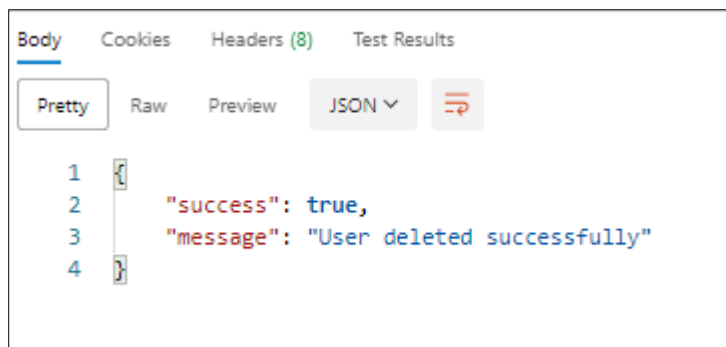
DELETE http://localhost:5000/users/68d2805a049e2af236c58403

Params Authorization Headers (9) Body Scripts Settings

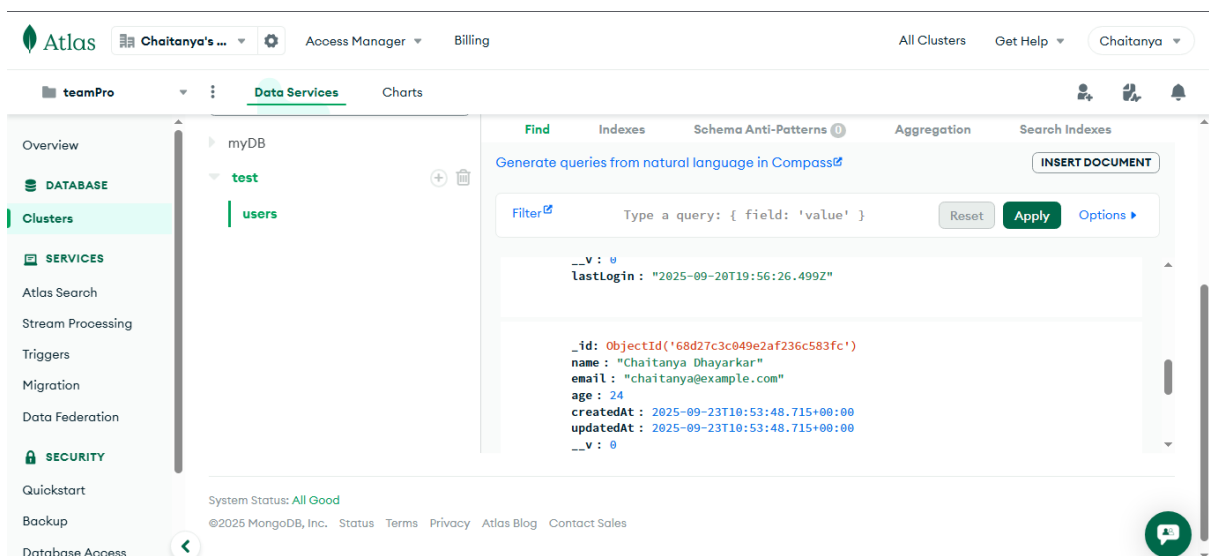
☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL JSON

```
1 {
2   "name": "Ram D.",
3   "email": "cd@example.com",
4   "age": 18
5 }
6
```

Response:



Mongo Atlas Server



30% Extra

2025-09-23T15:09:44.765872682Z ==> Cloning from <https://github.com/mr-chaitanyad/FSDLab4>

2025-09-23T15:09:46.241998963Z ==> Checking out commit
b7da3c394032ebb8f32162b29c6d512a6f39caa1 in branch master

2025-09-23T15:09:49.871637973Z ==> Using Node.js version 22.16.0 (default)

2025-09-23T15:09:49.89813843Z ==> Docs on specifying a Node.js version:
<https://render.com/docs/node-version>

2025-09-23T15:09:51.827464772Z ==> Running build command 'npm install'...

2025-09-23T15:09:53.415573968Z 2025-09-23T15:09:53.415600269Z added 2 packages, and
audited 446 packages in 1s 2025-09-23T15:09:53.41565489Z 2025-09-23T15:09:53.41567568Z 72
packages are looking for funding 2025-09-23T15:09:53.415703051Z run `npm fund` for details
2025-09-23T15:09:53.416654594Z found 0 vulnerabilities

2025-09-23T15:09:58.850245813Z ==> Uploading build...

2025-09-23T15:10:04.64482223Z ==> Uploaded in 4.2s. Compression took 1.6s

2025-09-23T15:10:04.660762269Z ==> Build successful 🍌

2025-09-23T15:10:10.991885199Z ==> Deploying...

2025-09-23T15:13:33.203460285Z ==> Running 'node index.js'

2025-09-23T15:13:33.500627609Z [dotenv@17.2.2] injecting env (0) from .env -- tip: ⚙️ enable debug logging with { debug: true }

2025-09-23T15:13:35.259645405Z ==> No open ports detected, continuing to scan...

2025-09-23T15:13:35.45512444Z ==> Docs on specifying a port: <https://render.com/docs/web-services#port-binding>

2025-09-23T15:13:36.494443485Z

mongodb+srv://chaitanyad:gAAXHhf6gN9FSA7B@cluster0.ebknbyr.mongodb.net/?retryWrites=true&w=majority&appName=Cluster0

2025-09-23T15:13:36.593975348Z 🚀 Server running at http://localhost:5000

2025-09-23T15:13:39.429670487Z ✅ MongoDB Connected

2025-09-23T15:13:41.574843277Z ==> Your service is live 🍌

2025-09-23T15:13:41.941432213Z ==> Available at your primary URL <https://fsdlab4.onrender.com>

Host on Render.com

SERVICE NAME	STATUS	RUNTIME	REGION	DEPLOYED
🌐 FSDLab4	✅ Deployed	Node	Oregon	5min

🌐 FSDLab4

🔍 Search

+ New

🔄 Upgrade

?

🔒

🔔 Your free instance will spin down with inactivity, which can delay requests by 50 seconds or more. [Upgrade now](#)

Sep 23, 2025 at 8:39 PM 🟢 Live
[b7da3c3](#) first commit

All logs

🔍 Search

🔄 Live tail

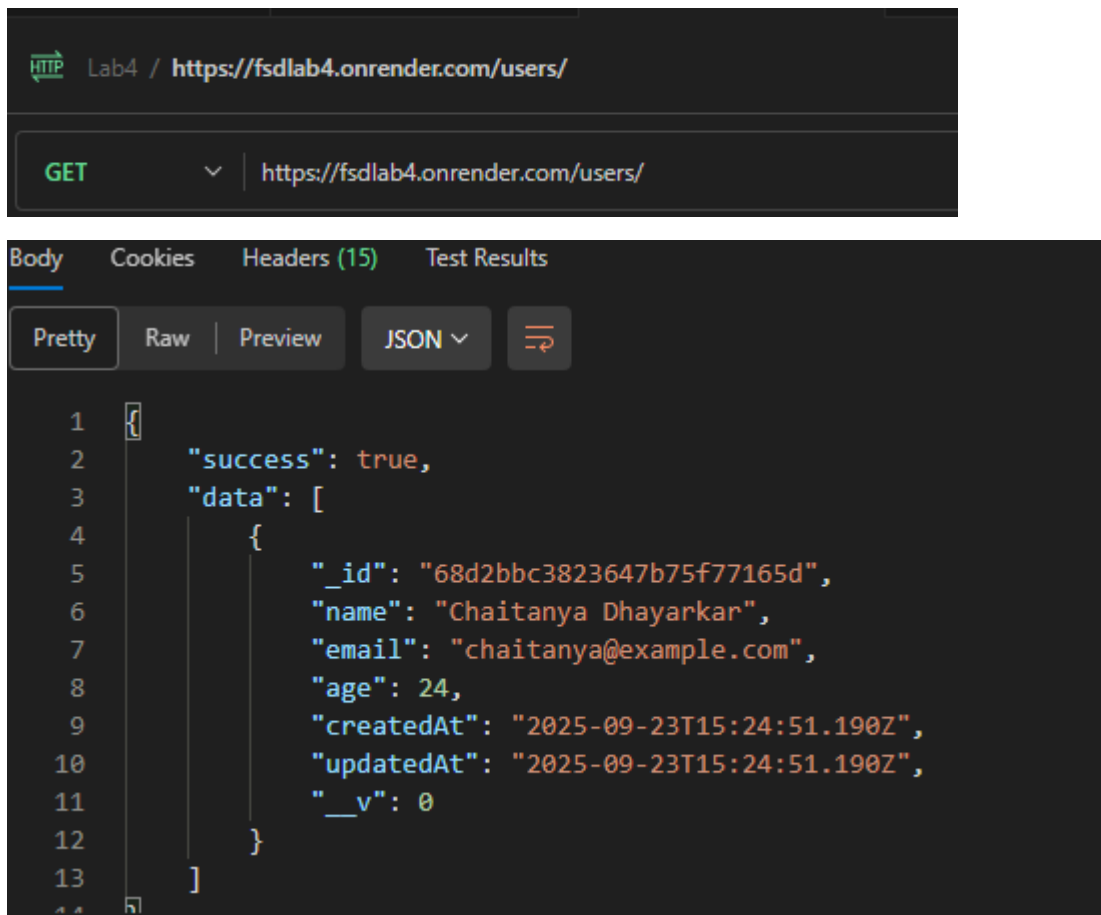
GMT+5:30

↕

...

Sep 23 08:39:44 PM 📌 --> Cloning from <https://github.com/mr-chaitanyad/FSDLab4>
Sep 23 08:39:46 PM 📌 --> Checking out commit [b7da3c394632ebb8f32162b29c6d512a6f39caa1](#) in branch master
Sep 23 08:39:49 PM 📌 --> Using Node.js version 22.10.0 (default)
Sep 23 08:39:49 PM 📌 --> Docs on specifying a Node.js version: <https://render.com/docs/node-version>
Sep 23 08:39:51 PM 📌 --> Running build command 'npm install'...
Sep 23 08:39:53 PM 📌
Sep 23 08:39:53 PM 📌 added 2 packages, and audited 446 packages in 1s
Sep 23 08:39:53 PM 📌
Sep 23 08:39:53 PM 📌 72 packages are looking for funding
Sep 23 08:39:53 PM 📌 run 'npm fund' for details
Sep 23 08:39:53 PM 📌
Sep 23 08:39:53 PM 📌 found 0 vulnerabilities
Sep 23 08:39:58 PM 📌 --> Uploading build...

Check Deployed website API working



Conclusion:

We have successfully implemented and execute REST API Design with MongoDB + Mongoose Integration