**Lab Assignment No. 3 B**

**Name: Usha sree alekhya siddi**

**Class: TE IT A**

**Roll No.: 35071**

**Subject: Web Application Development**

**Title: Create four API using Node.JS, ExpressJS and MongoDB for CURD Operations on assignment 2.C.**

**server.js**

// import packages

const express = require('express');

const cors = require('cors');

const mongoose = require('mongoose');

// import router from routes

const userRoute = require('./route/userRoute');

// initialize app

const app = express();

app.use(express.json());

app.use(cors());

// start point of api

app.use('/api', userRoute);

//

mongoose

.connect('mongodb://0.0.0.0:27017/puneuniversity')

.then(() => {

app.listen(4000, () => {

console.log('Database is connected and server is running on port 4000!');

});

})

.catch((error) => {

console.log(error);

});

**userRoute.js**

const express = require('express');

// initialize router from express

const router = express.Router();

// import all functions from userController

const {

register,

login,

updateUser,

deleteUser,

getUser,

} = require('../controllers/userController');

// all api routes

// @Method --> POST @Path --> http://localhost:4000/api/register

router.post('/register', register);

// @Method --> POST @Path --> http://localhost:4000/api/login

router.post('/login', login);

// @Method --> PUT @Path --> http://localhost:4000/api/updateuser

router.put('/updateuser', updateUser);

// @Method --> DELETE @Path --> http://localhost:4000/api/deleteuser:email

router.delete('/deleteuser/:email', deleteUser);

// @Method --> GET @Path --> http://localhost:4000/api/getuser:email

router.get('/getuser/:email', getUser);

// export the router

module.exports = router;

**userController.js**

// import user model from model

const User = require('../model/userModel');

// register user

const register = async (req, res) => {

try {

// create new user with req.body as data

const user = await User.create(req.body);

// if user get created successfully then return success and user data

if (user) {

res.json({ success: true, data: user });

}

// else return success false

else {

res.json({ success: false, data: 'Cannot register user' });

}

} catch (error) {

console.log(error);

}

};

// login user

const login = async (req, res) => {

try {

// find the user by email and password

const user = await User.findOne({

email: req.body.email,

password: req.body.password,

});

// if user exist then return success and user data

if (user) {

res.json({ success: true, data: user });

}

// else return success false

else {

res.json({

success: false,

data: 'Please enter valid email and password',

});

}

} catch (error) {

console.log(error);

}

};

const updateUser = async (req, res) => {

try {

// find the user by email and update the data provided in req.body

const user = await User.findOneAndUpdate(

{ email: req.body.email },

req.body,

{ new: true }

);

// if user get updated then return data

if (user) {

res.json({ success: true, data: user });

}

// else return success false

else {

res.json({

success: false,

data: 'Cannot update user',

});

}

} catch (error) {

console.log(error);

}

};

const deleteUser = async (req, res) => {

try {

// delete user based on email id given in req parameter --> /deleteuser/email@gmai.com

const user = await User.findOneAndDelete({ email: req.params.email });

// if user get deleted then return data of deleted user

if (user) {

res.json({ success: true, data: user });

}

// else return success false

else {

res.json({

success: false,

data: 'Cannot delete user',

});

}

} catch (error) {

console.log(error);

}

};

const getUser = async (req, res) => {

try {

// find user based on email id given in req parameter --> /deleteuser/email@gmai.com

const user = await User.findOne({ email: req.params.email });

// if user exist then return user data

if (user) {

res.json({ success: true, data: user });

}

// else return success false

else {

res.json({

success: false,

data: 'User not found',

});

}

} catch (error) {

console.log(error);

}

};

module.exports = {

register,

login,

updateUser,

deleteUser,

getUser,

};

**userModel.js**

const mongoose = require('mongoose');

// creating the mongoose schema -> schema defines structure of your data in JSON format

const userSchema = mongoose.Schema({

name: { type: String, required: true },

email: { type: String, required: true },

password: { type: String, required: true },

gender: { type: String, required: true },

branch: { type: String, required: true },

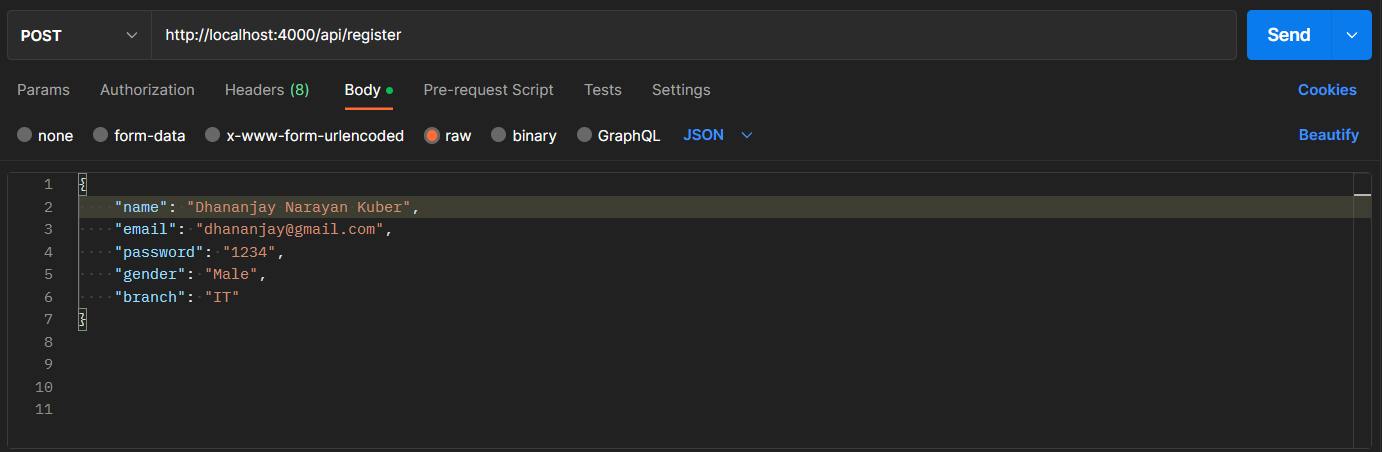
});

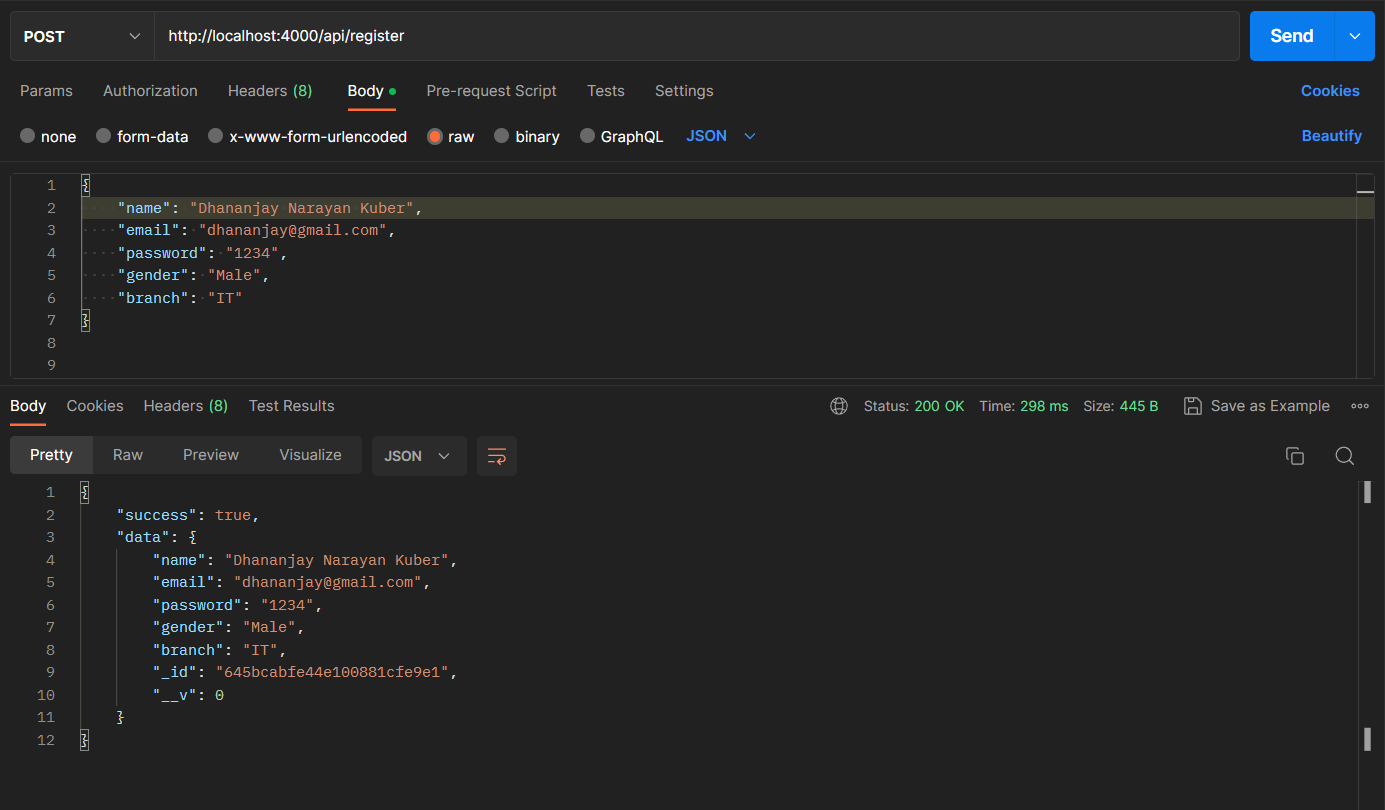
// using schema we are creating database model called user (user is nothing but the name for collection)

module.exports = mongoose.model('user', userSchema);

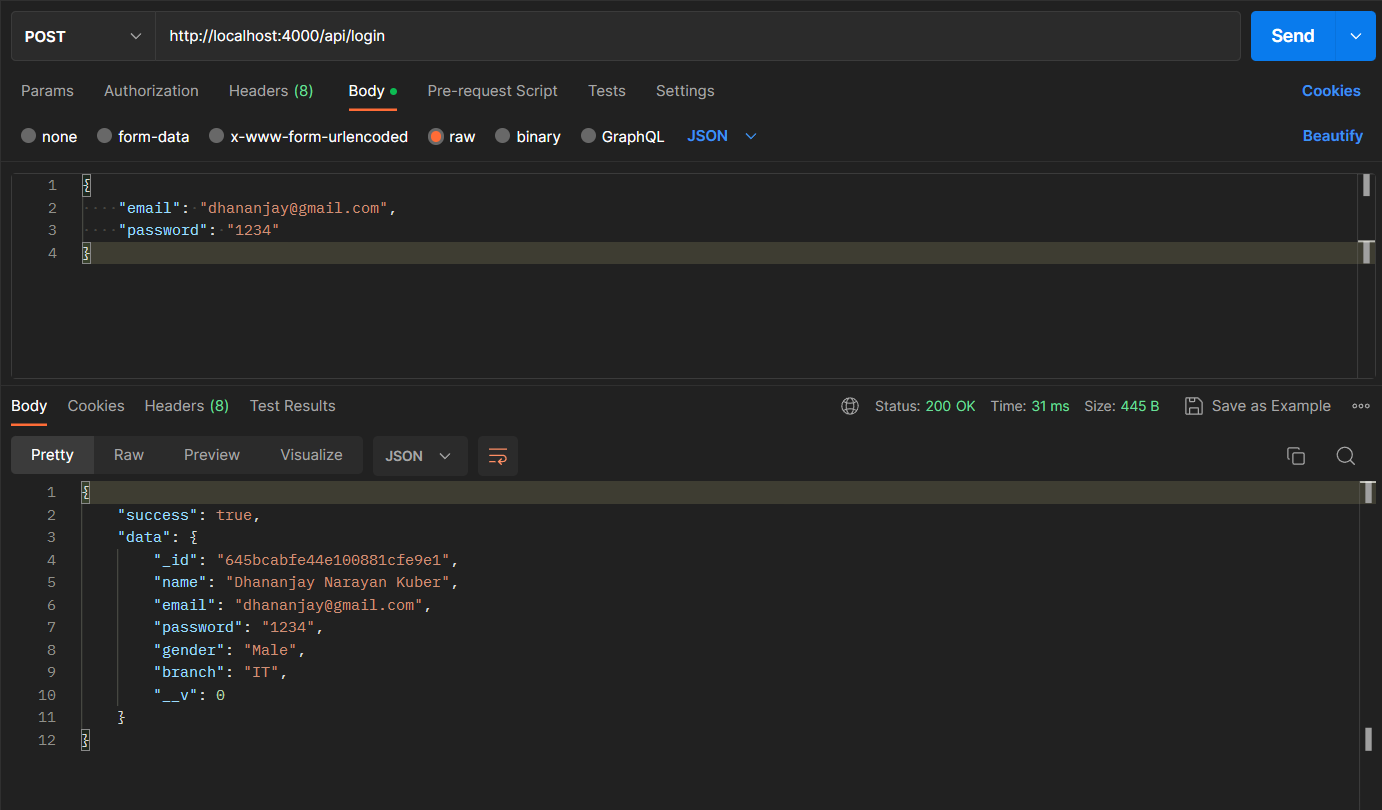
**Output**

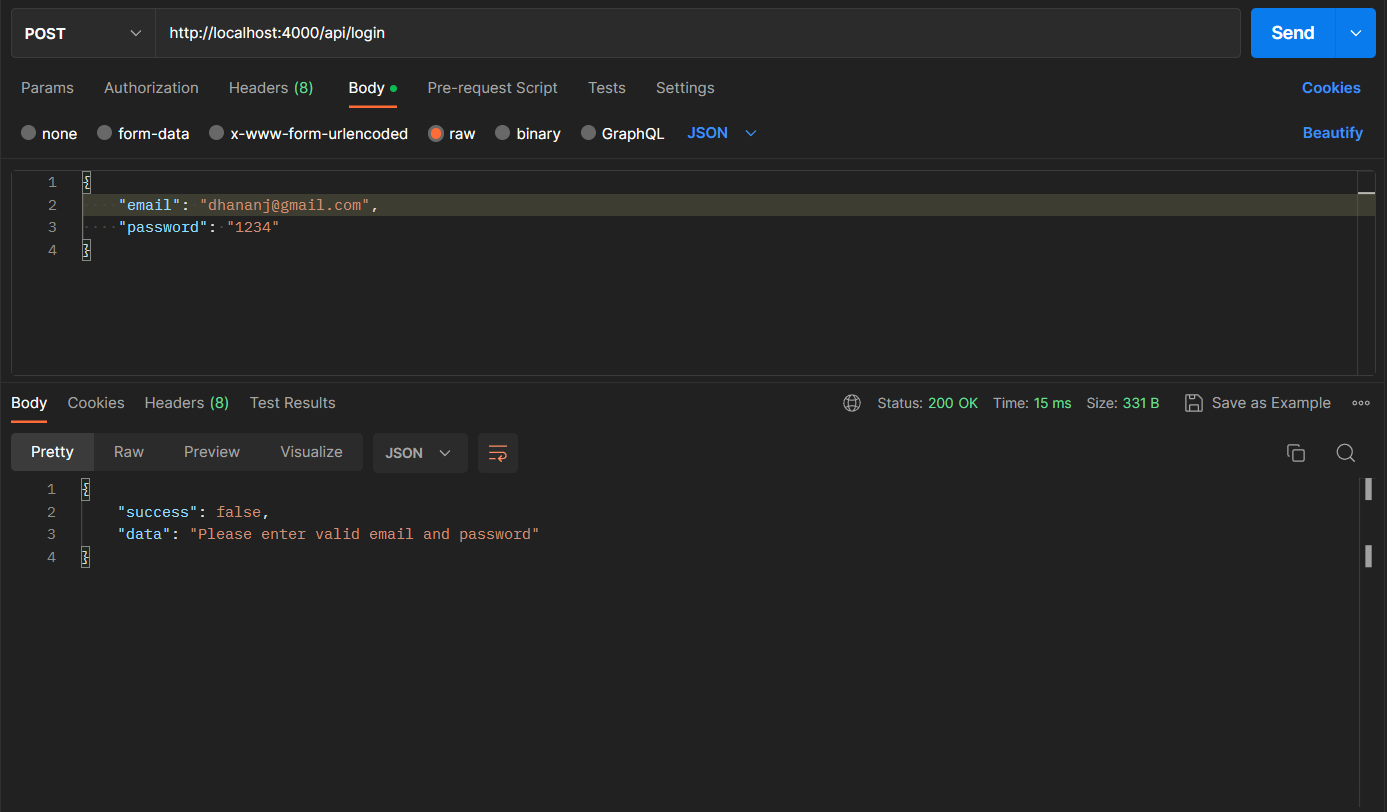
1. **Register**

****

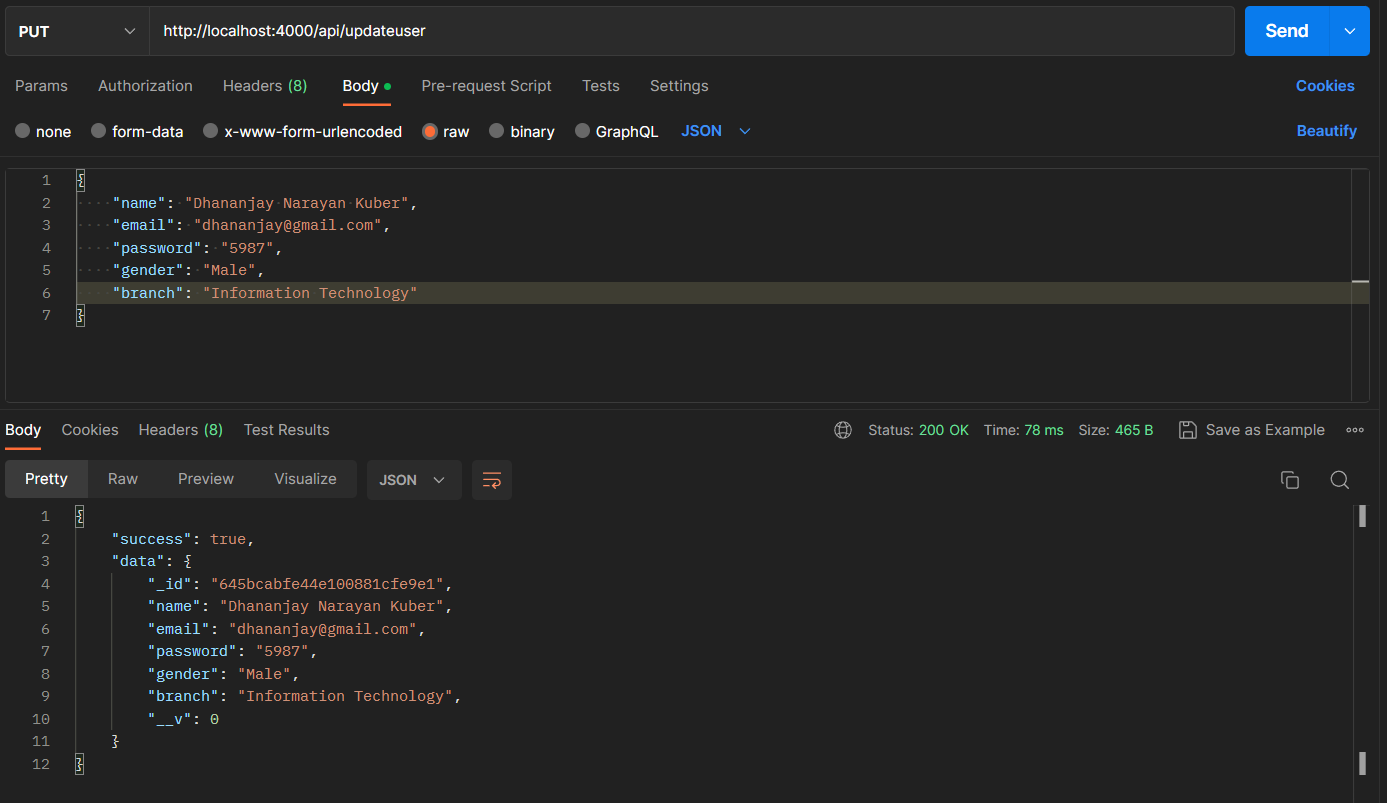
****

1. **Login**

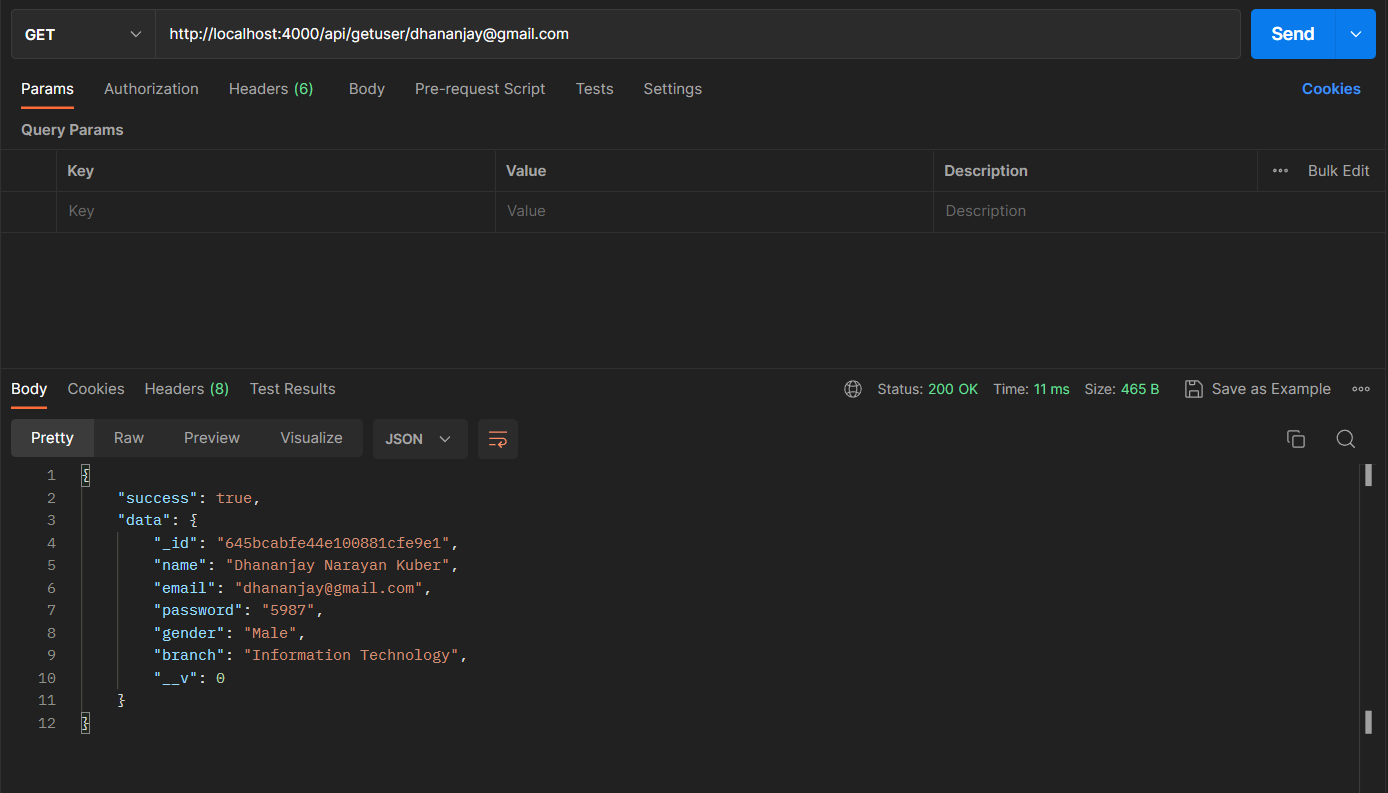




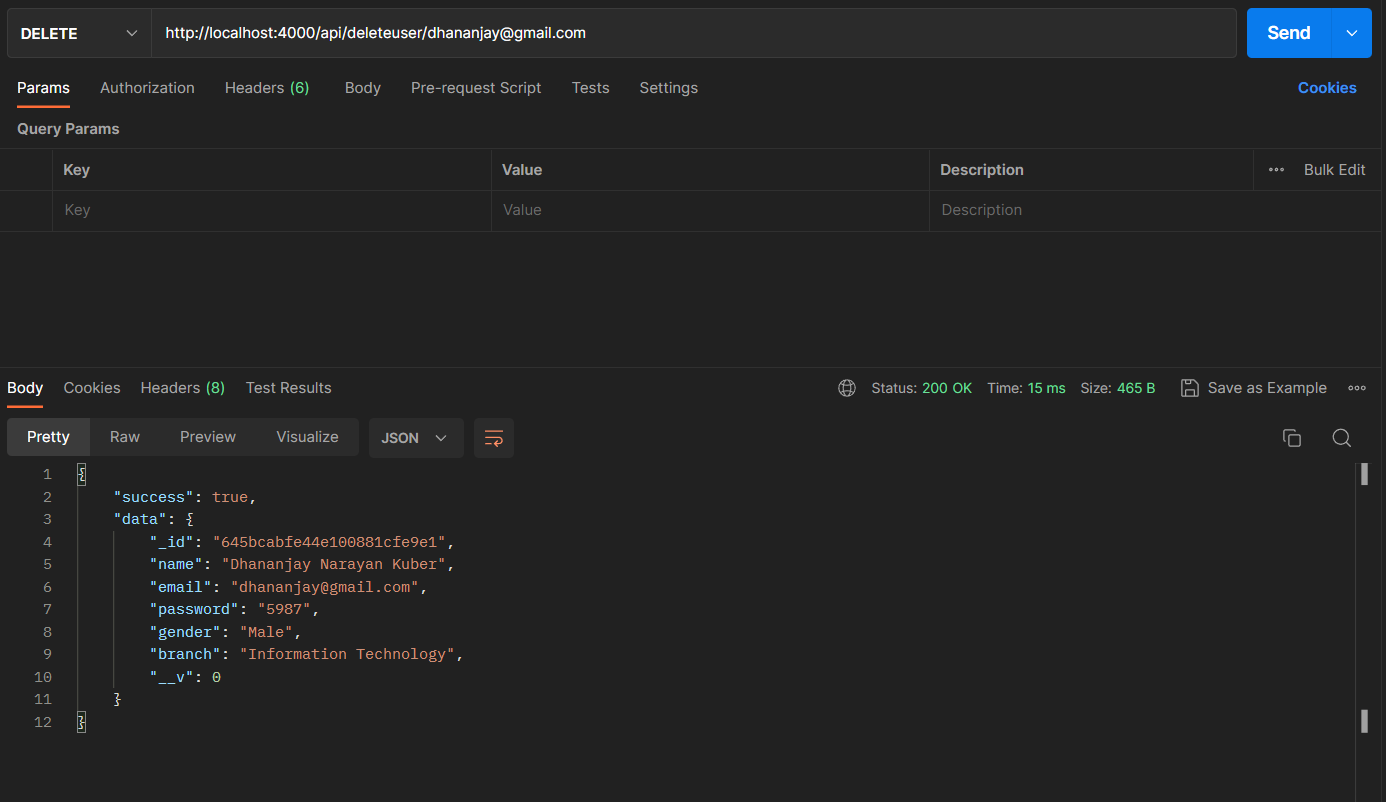
1. **Update**

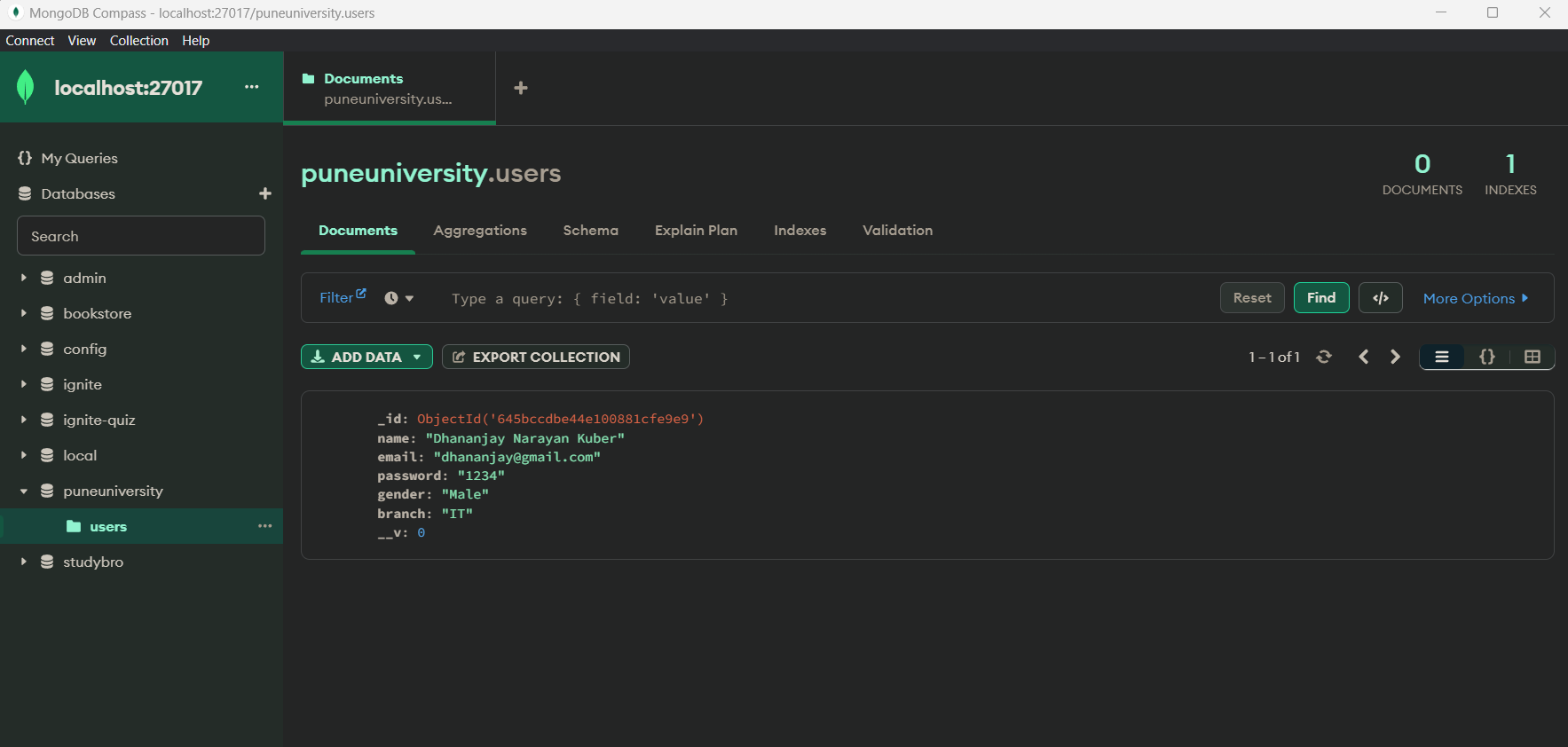
****

1. **Get**

****

1. **Delete**

****

****