

```
1 var express = require("express");
2 var router = express.Router();
3 const userModel = require("./users");
4 const postModel = require("./post");
5
6 /* GET home page. */
7 router.get("/", function (req, res, next) {
8   res.render("index", { title: "Data Association" });
9 });
10
11 router.get("/alluserposts", async function (req, res, next) {
12   let user = await userModel
13     .findOne({ _id: "658ae7c43c020433382a02bc" })
14     .populate("posts");
15   res.send(user);
16 });
17
18 router.get("/createuser", async function (req, res, next) {
19   let createduser = await userModel.create({
20     username: "ajitaa",
21     password: "123456",
22     posts: [],
23     email: "ajitkumar@gmail.com",
24     fullName: "Ajit Kumaraaa",
25   });
26   res.send(createduser);
27 });
28
29 router.get("/createpost", async function (req, res, next) {
30   let createdpost = await postModel.create({
31     postText: "my name is Ajit kumar",
32     user: "658ae7c43c020433382a02bc",
33   });
34   let user = await userModel.findOne({ _id: "658ae7c43c020433382a02bc" });
35   user.posts.push(createdpost._id);
36   await user.save();
37   res.send("done");
38 });
39 module.exports = router;
40
41
42 //first model create
43 const mongoose = require("mongoose");
44
45 mongoose.connect("mongodb://127.0.0.1:27017/dataassociation");
46
47 const userSchema = new mongoose.Schema({
48   username: {
49     type: String,
50     required: true,
51     unique: true,
52   },
53   password: {
54     type: String,
55     required: true,
56   },
57   posts: [
58     {
59       type: mongoose.Schema.Types.ObjectId,
60       ref: "Post",
```

```
61     },
62   ],
63   dp: {
64     type: String,
65   },
66   email: {
67     type: String,
68     required: true,
69     unique: true,
70   },
71   fullName: {
72     type: String,
73     required: true,
74   },
75 });
76 module.exports = mongoose.model("User", userSchema);
77
78 //second model create
79 const mongoose = require("mongoose");
80
81 const postSchema = new mongoose.Schema({
82   postText: {
83     type: String,
84     required: true,
85   },
86   user: {
87     type: mongoose.Schema.Types.ObjectId,
88     ref: "User",
89   },
90   createdAt: {
91     type: Date,
92     default: Date.now,
93   },
94   likes: {
95     type: Array,
96     default: [],
97   },
98 });
99
100 module.exports = mongoose.model("Post", postSchema);
101
102
103
104
```