

## Assignment 3B

Code :

### **Index: server.js**

```
const express = require("express");
const bodyParser = require("body-parser");
const app = express();

app.use(bodyParser.urlencoded(
{ extended: true }));

app.use(bodyParser.json());
const UserRoute = require("./routes/User");
app.use("/user", UserRoute);

const dbConfig = require("./config/database.config.js"); const mongoose =
require("mongoose");
mongoose.Promise = global.Promise; mongoose .connect(dbConfig.url, { useNewUrlParser:
true,
})
.then(() => {console.log("Database Connected Successfully!!");
})
.catch((err) => { console.log("Could not connect to the database", err);
process.exit(); });
app.get("/", (req, res) => {
    res.json({ message: "Hello
Crud Node Express" }); });

app.listen(3000, () => { console.log("Server is
listening on port 3000");
```

### **Routes: User.js**

```
const express = require("express");
const UserController = require("../controllers/User")
const router = express.Router();
router.get("/", UserController.findAll); router.get("/:id", UserController.findOne);
router.post("/", UserController.create); router.patch("/:id", UserController.update);
router.delete("/:id", UserController.destroy);

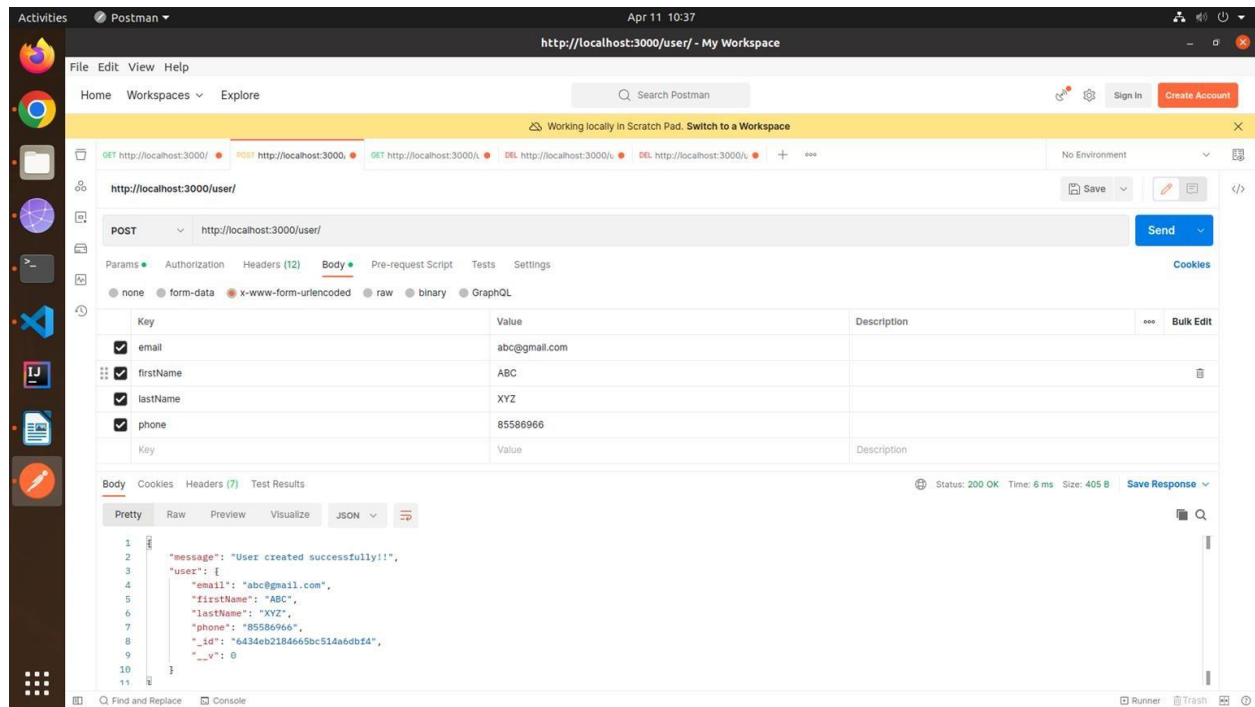
module.exports = router;
```

### **Model: User.js**

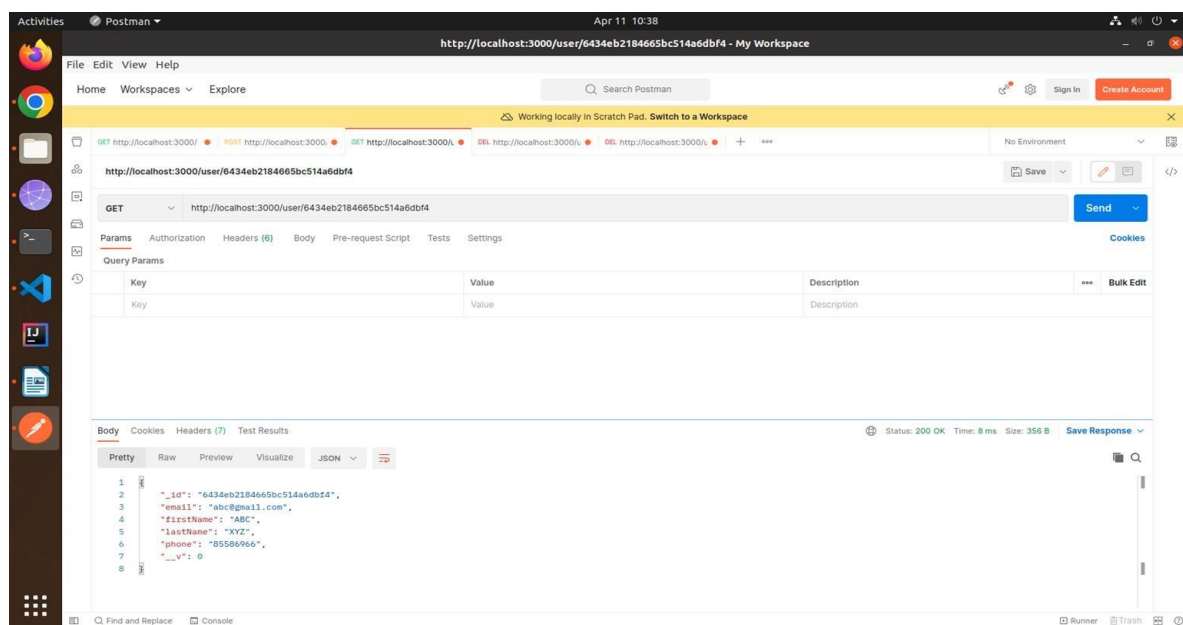
```
var mongoose = require("mongoose");
var schema = new mongoose.Schema({
  email: { type: String, required: true, unique: true },
  firstName: { type: String, default: "" },
  lastName: { type: String, default: "" },
  phone: String,
});
var user = new mongoose.model("User", schema);
module.exports = user;
```

Output :

Creating user



Displaying single user



## Updating user

A screenshot of the Postman application interface. The top bar shows the URL `http://localhost:3000/user/6434eb2184665bc514a6dbf4` and the workspace name "My Workspace". The main area displays a GET request to the same URL. The "Params" tab is active, showing a table with two columns: "Key" and "Value". The "Body" tab is also visible, showing a JSON response with user details. The status bar at the bottom indicates "Status: 200 OK" and "Time: 8 ms".

| Key | Value | Description |
|-----|-------|-------------|
| Key | Value | Description |

```
{  "id": "6434eb2184665bc514a6dbf4",  "email": "abc@gmail.com",  "firstName": "ABC",  "lastName": "XYZ",  "phone": "85586966",  "v": 0}
```

## Deleting user

A screenshot of the Postman application interface. The top bar shows the URL `http://localhost:3000/user/6434eb2184665bc514a6dbf4` and the workspace name "My Workspace". The main area displays a DELETE request to the same URL. The "Params" tab is active, showing a table with two columns: "Key" and "Value". The "Body" tab is also visible, showing a JSON response with a success message. The status bar at the bottom indicates "Status: 200 OK" and "Time: 5 ms".

| Key | Value | Description |
|-----|-------|-------------|
| Key | Value | Description |

```
{  "message": "User deleted successfully!"}
```