

## 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.con
fig.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:

## Create User:

The screenshot shows the Postman application interface. The URL bar displays `http://localhost:3000/user/ - My Workspace`. The request method is set to **POST** and the URL is `http://localhost:3000/user/`. The request body is formatted as **x-www-form-urlencoded** and contains the following data:

Key	Value	Description
email	abc@gmail.com	
firstName	ABC	
lastName	XYZ	
phone	85586966	

The response status is **200 OK** with a time of 6 ms and a size of 405 B. The response body is displayed in the **Body** tab, showing a JSON object:

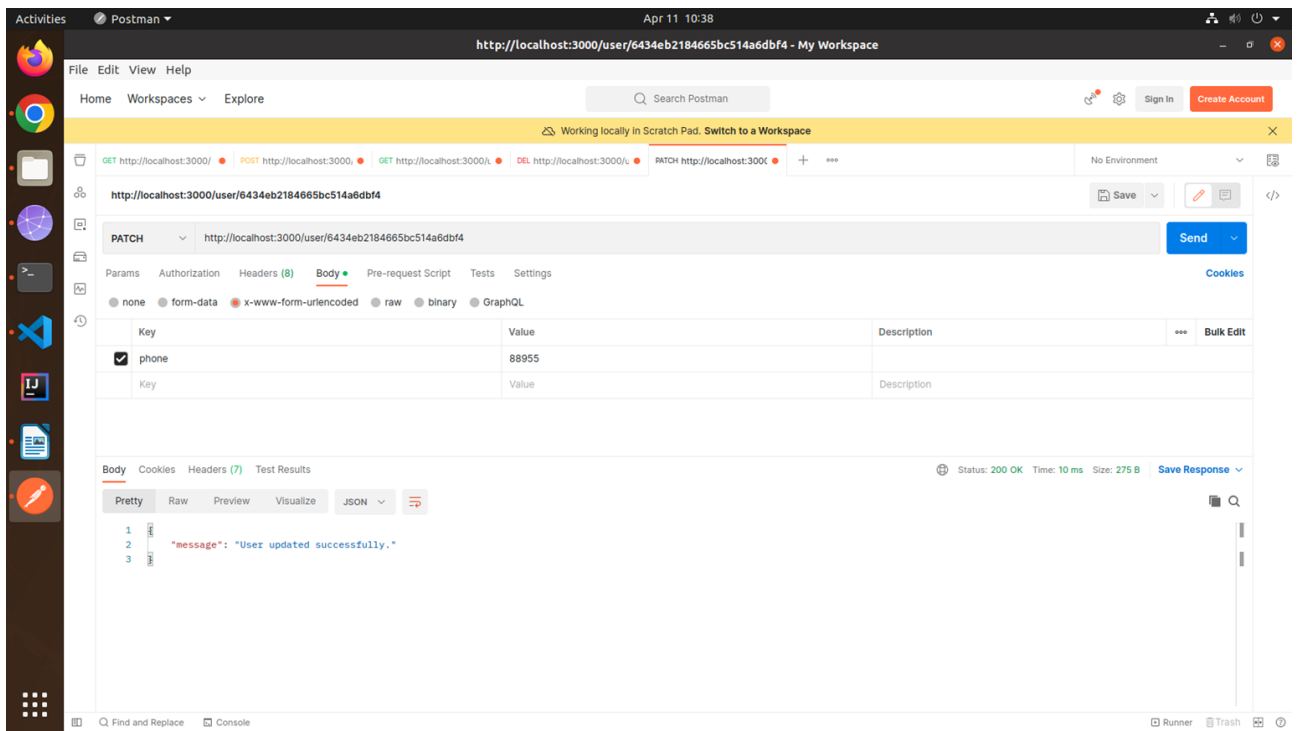
```
1 {
2   "message": "User created successfully!!",
3   "user": {
4     "email": "abc@gmail.com",
5     "firstName": "ABC",
6     "lastName": "XYZ",
7     "phone": "85586966",
8     "_id": "6434eb2184665bc514a6dbf4",
9     "__v": 0
10  }
11 }
```

## Display Single User:

The screenshot shows the Postman application interface. The URL bar displays `http://localhost:3000/user/6434eb2184665bc514a6dbf4 - My Workspace`. The request method is set to **GET** and the URL is `http://localhost:3000/user/6434eb2184665bc514a6dbf4`. The response status is **200 OK** with a time of 8 ms and a size of 356 B. The response body is displayed in the **Body** tab, showing a JSON object:

```
1 {
2   "_id": "6434eb2184665bc514a6dbf4",
3   "email": "abc@gmail.com",
4   "firstName": "ABC",
5   "lastName": "XYZ",
6   "phone": "85586966",
7   "__v": 0
8 }
```

## Update User:



## Delete User:

