

### ASSIGNMENT-3

#### FULL STACK DEVELOPMENT (MERN STACK)

REGISTER NUMBER :SBAP0008297  
NAME :Perugu Hinduja  
G-MAIL :peruguhindujareddy@gmail.com  
ROLL NUMBER :209F1A0536  
COURSE :Full Stack Development (MERN stack).  
COLLEGE :Sri Venkateswara Ins tute of Technology, Anantapur.

#### server.js

```
const express = require('express');
const cors = require("cors");
const bodyParser = require('body-parser');
const app = express();
app.use(bodyParser.urlencoded({ extended: true }))
app.use(bodyParser.json())
app.use(cors());
const dbConfig = require('./config/database.config.js');
const mongoose = require('mongoose');
mongoose.Promise = global.Promise;
mongoose.connect(dbConfig.url, {
  useNewUrlParser: true
}).then(() => {
  console.log("Databse 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(4000, () => {
  console.log("Server is listening on port 4000");
});
const UserRoute = require('./app/routes/user')
app.use('/user', UserRoute)
```

#### modeluser.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;

```

controllersUser.js

```

const UserModel = require('../model/user')
// Create and Save a new user
exports.create = async (req, res) => {
  if (!req.body.email && !req.body.firstName && !req.body.lastName && !req.body.phone) {
    res.status(400).send({ message: "Content can not be empty!" });
  }

  const user = new UserModel({
    email: req.body.email,
    firstName: req.body.firstName,
    lastName: req.body.lastName,
    phone: req.body.phone
  });

  await user.save().then(data => {
    res.send({
      message: "User created successfully!!",
      user: data
    });
  }).catch(err => {
    res.status(500).send({

```

```

        message: err.message || "Some error occurred while creating user"
    });
});
};
// Retrieve all users from the database.
exports.findAll = async (req, res) => {
    try {
        const user = await UserModel.find();
        res.status(200).json(user);
    } catch(error) {
        res.status(404).json({message: error.message});
    }
};
// Find a single User with an id
exports.findOne = async (req, res) => {
    try {
        const user = await UserModel.findById(req.params.id);
        res.status(200).json(user);
    } catch(error) {
        res.status(404).json({ message: error.message});
    }
};
// Update a user by the id in the request
exports.update = async (req, res) => {
    if(!req.body) {
        res.status(400).send({
            message: "Data to update can not be empty!"
        });
    }

    const id = req.params.id;

    await UserModel.findByIdAndUpdate(id, req.body, { useFindAndModify: false }).then(data => {
        if (!data) {
            res.status(404).send({
                message: `User not found.`
            });
        } else {
            res.send({ message: "User updated successfully." })
        }
    }).catch(err => {
        res.status(500).send({
            message: err.message
        });
    });

```

```

    });
  };
  // Delete a user with the specified id in the request
  exports.destroy = async (req, res) => {
    await UserModel.findByIdAndDelete(req.params.id).then(data => {
      if (!data) {
        res.status(404).send({
          message: `User not found.`
        });
      } else {
        res.send({
          message: "User deleted successfully!"
        });
      }
    }).catch(err => {
      res.status(500).send({
        message: err.message
      });
    });
  };
};

```

### routesUser.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

```







