Name: Dominic Gopalakrishnan

ID: 101289239

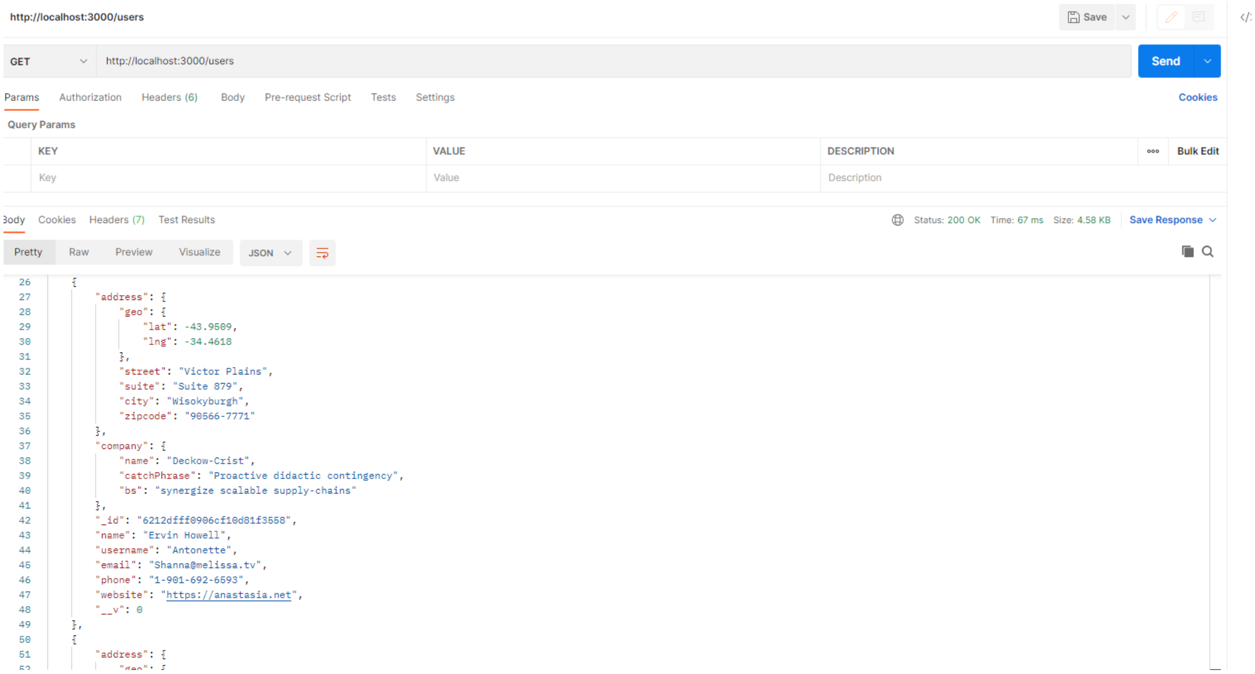
CRN: 49746

Full Stack Development

Lab 4 – Mongoose Validation

Screenshots:

Postman get:



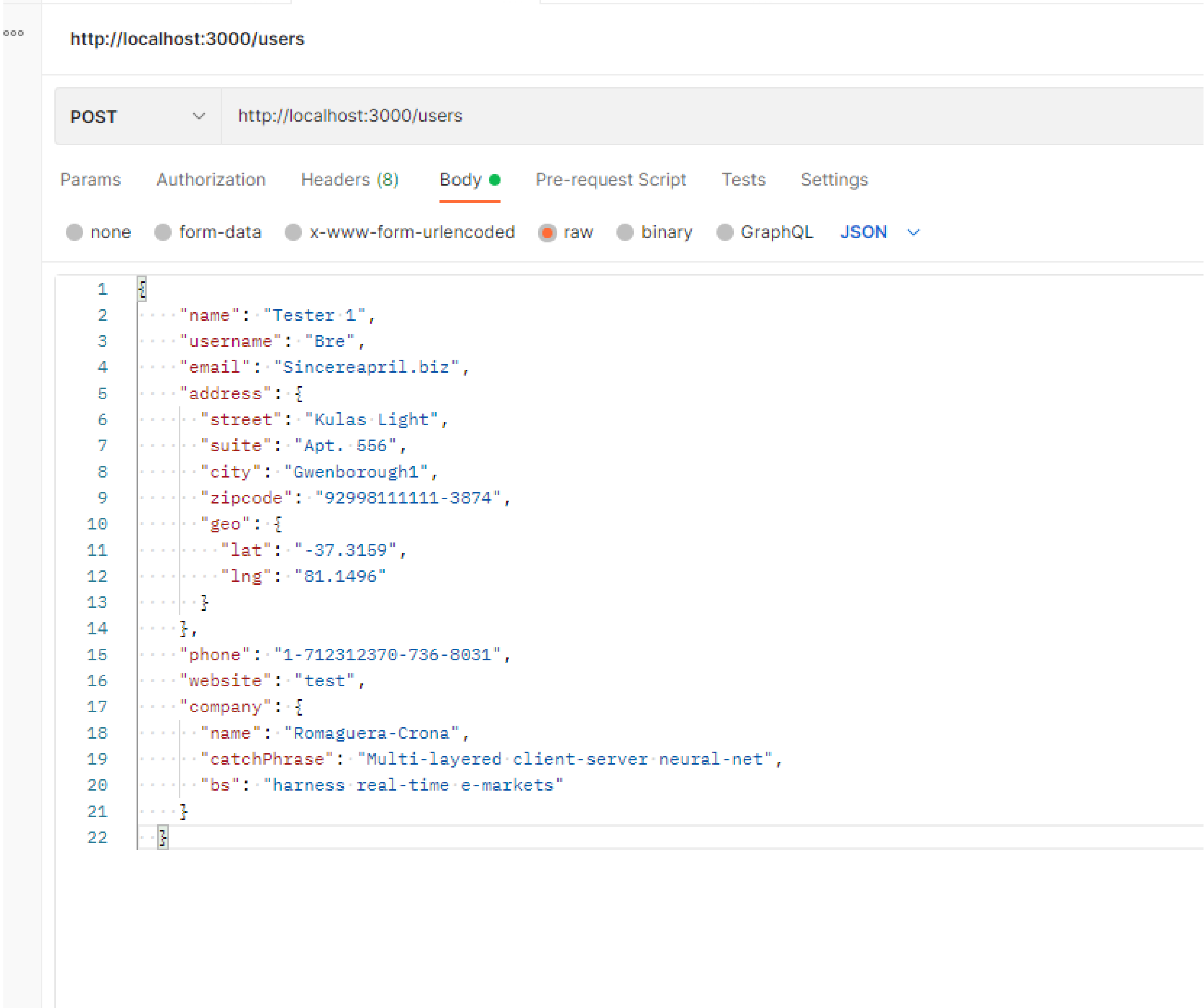
http://localhost:3000/users

GET http://localhost:3000/users

Status: 200 OK Time: 67 ms Size: 4.58 KB

```
    {
        "address": {
            "geo": {
                "lat": -43.9509,
                "lng": -34.4618
            },
            "street": "Victor Plains",
            "suite": "Suite 879",
            "city": "Wisokyburgh",
            "zipcode": "90566-7771"
        },
        "company": {
            "name": "Deckow-Crist",
            "catchPhrase": "Proactive didactic contingency",
            "bs": "synergize scalable supply-chains"
        },
        "_id": "6212dfff0906cf10d81f3558",
        "name": "Ervin Howell",
        "username": "Antonette",
        "email": "Shanna@melissa.tv",
        "phone": "1-901-692-6593",
        "website": "https://anastasia.net",
        "__v": 0
    },
    {
        "address": {
            "geo": {
```

Error filled postman body:

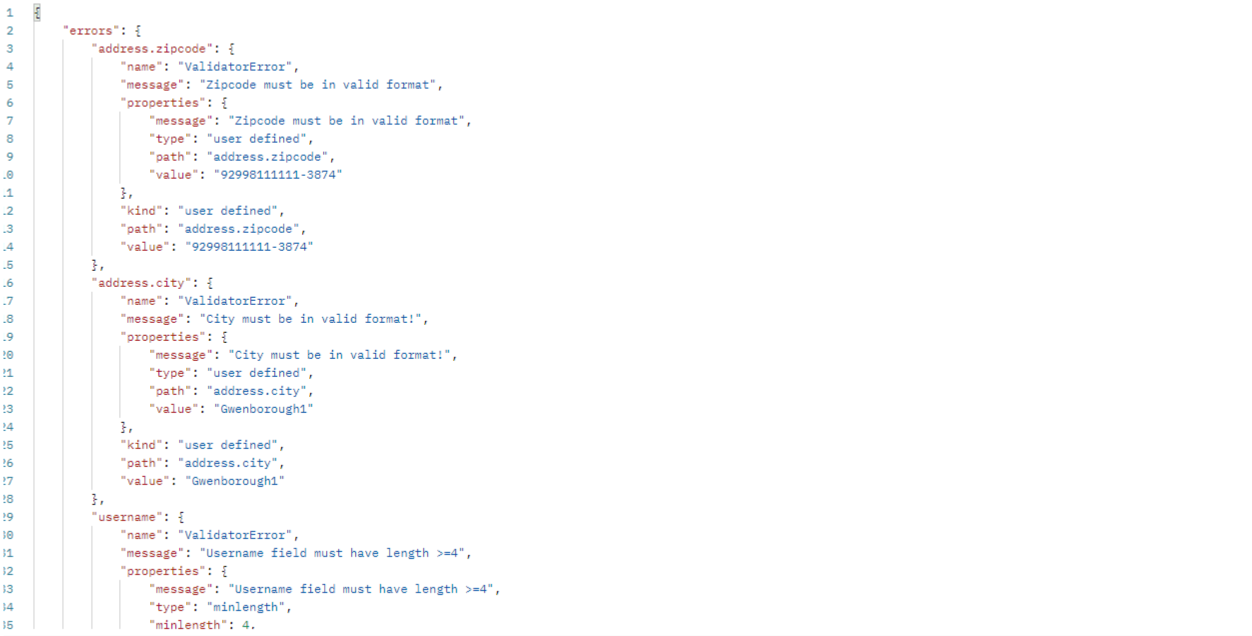


http://localhost:3000/users

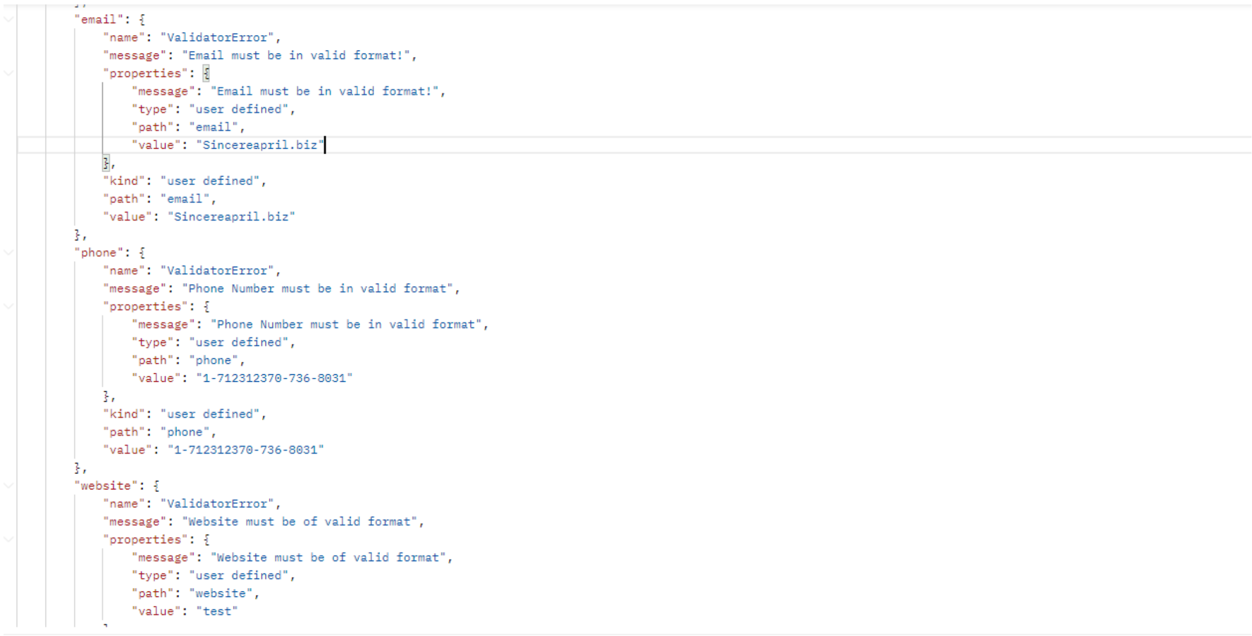
POST http://localhost:3000/users

```
{
    "name": "Tester 1",
    "username": "Bre",
    "email": "Sincereapril.biz",
    "address": {
      "street": "Kulas Light",
      "suite": "Apt. 556",
      "city": "Gwenborough1",
      "zipcode": "92998111111-3874",
      "geo": {
        "lat": "-37.3159",
        "lng": "81.1496"
      }
    },
    "phone": "1-712312370-736-8031",
    "website": "test",
    "company": {
      "name": "Romaguera-Crona",
      "catchPhrase": "Multi-layered client-server neural-net",
      "bs": "harness real-time e-markets"
    }
  }
```

Results/error validation:

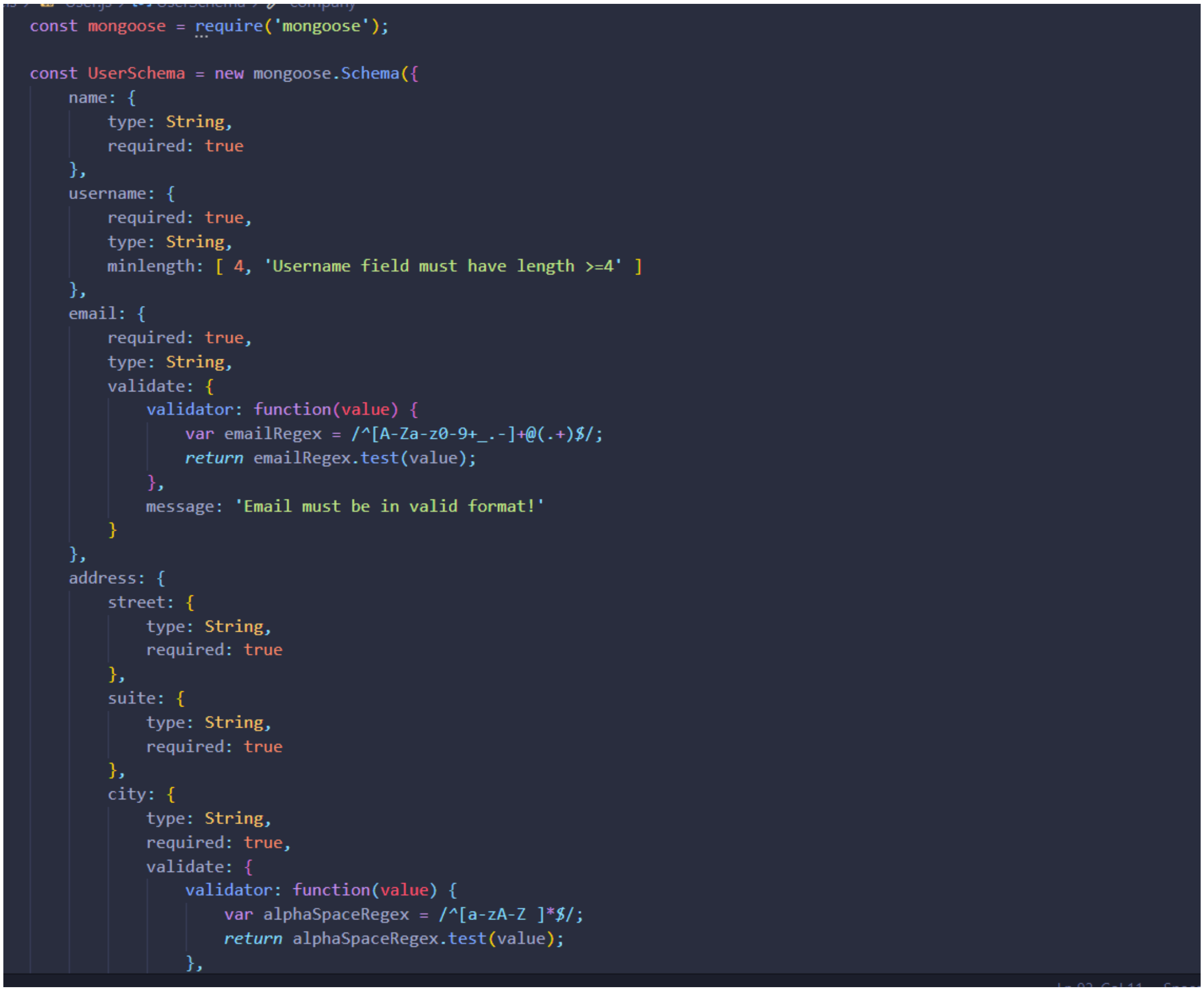


```
{
    "errors": {
        "address.zipcode": {
            "name": "ValidatorError",
            "message": "Zipcode must be in valid format",
            "properties": {
                "message": "Zipcode must be in valid format",
                "type": "user defined",
                "path": "address.zipcode",
                "value": "92998111111-3874"
            },
            "kind": "user defined",
            "path": "address.zipcode",
            "value": "92998111111-3874"
        },
        "address.city": {
            "name": "ValidatorError",
            "message": "City must be in valid format!",
            "properties": {
                "message": "City must be in valid format!",
                "type": "user defined",
                "path": "address.city",
                "value": "Gwenborough1"
            },
            "kind": "user defined",
            "path": "address.city",
            "value": "Gwenborough1"
        },
        "username": {
            "name": "ValidatorError",
            "message": "Username field must have length >=4",
            "properties": {
                "message": "Username field must have length >=4",
                "type": "minlength",
                "minlength": 4,
```



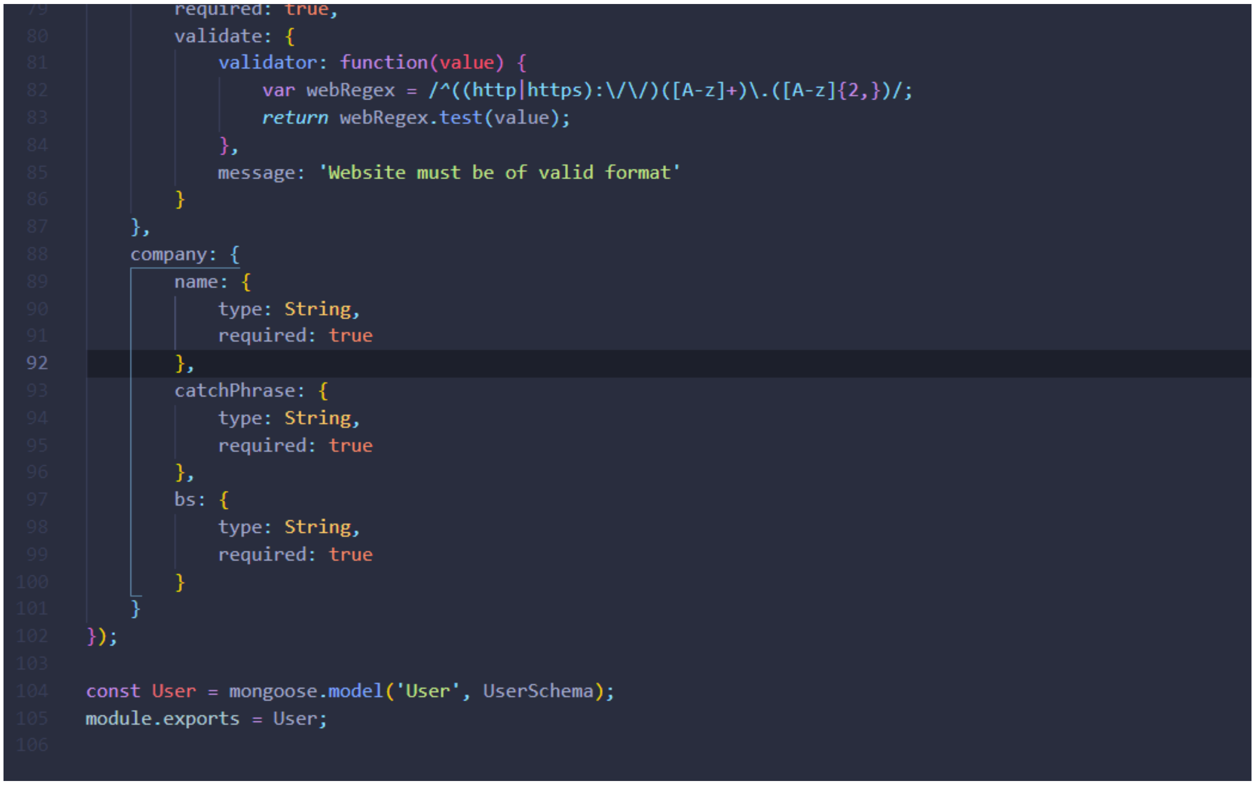
```
        "email": {
            "name": "ValidatorError",
            "message": "Email must be in valid format!",
            "properties": {
                "message": "Email must be in valid format!",
                "type": "user defined",
                "path": "email",
                "value": "Sincereapril.biz"
            },
            "kind": "user defined",
            "path": "email",
            "value": "Sincereapril.biz"
        },
        "phone": {
            "name": "ValidatorError",
            "message": "Phone Number must be in valid format",
            "properties": {
                "message": "Phone Number must be in valid format",
                "type": "user defined",
                "path": "phone",
                "value": "1-712312370-736-8031"
            },
            "kind": "user defined",
            "path": "phone",
            "value": "1-712312370-736-8031"
        },
        "website": {
            "name": "ValidatorError",
            "message": "Website must be of valid format",
            "properties": {
                "message": "Website must be of valid format",
                "type": "user defined",
                "path": "website",
                "value": "test"
```

User Schema:



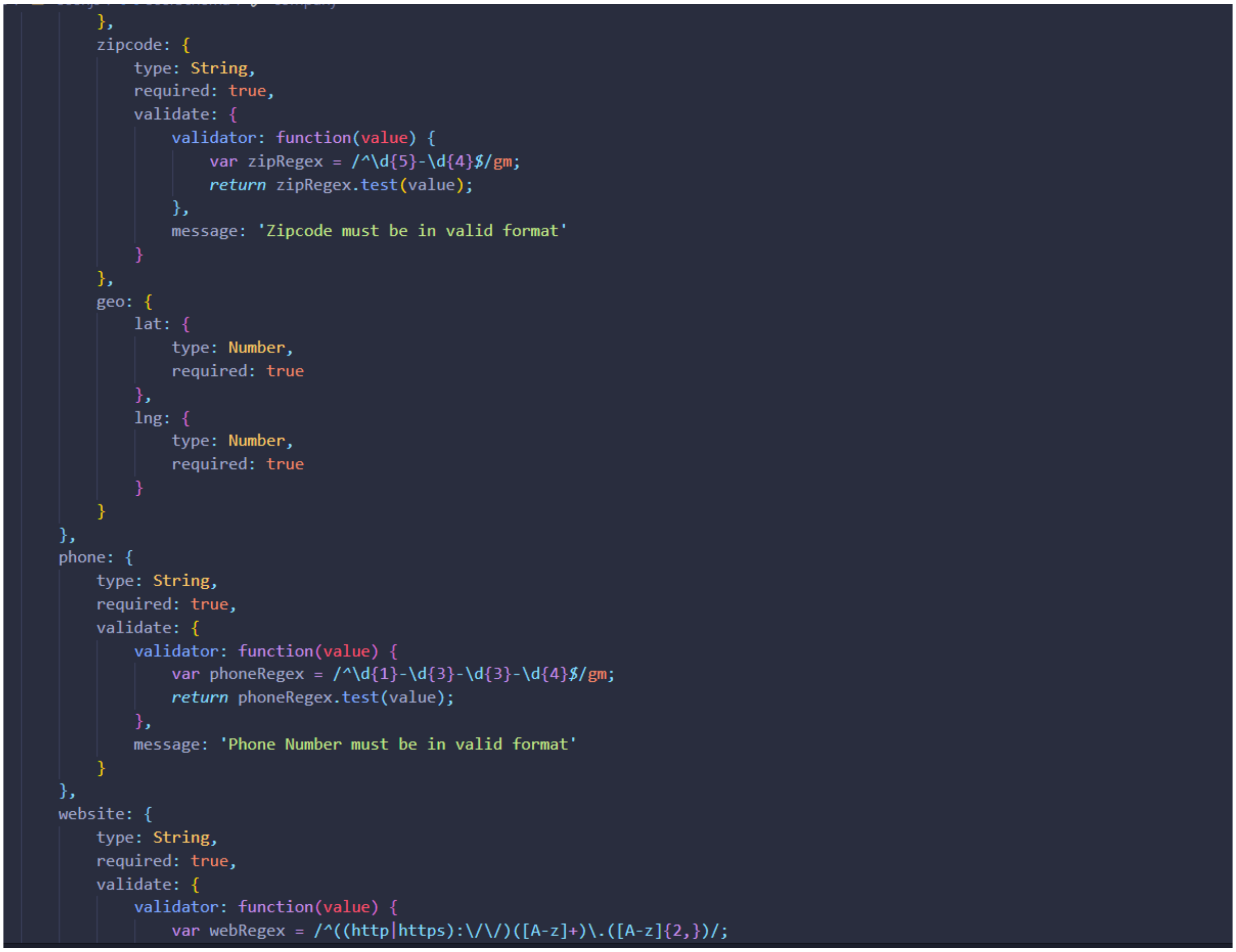
```
const mongoose = require('mongoose');

const UserSchema = new mongoose.Schema({
    name: {
        type: String,
        required: true
    },
    username: {
        required: true,
        type: String,
        minlength: [ 4, 'Username field must have length >=4' ]
    },
    email: {
        required: true,
        type: String,
        validate: {
            validator: function(value) {
                var emailRegex = /^[A-Za-z0-9+_.-]+@(.+)$/;
                return emailRegex.test(value);
            },
            message: 'Email must be in valid format!'
        }
    },
    address: {
        street: {
            type: String,
            required: true
        },
        suite: {
            type: String,
            required: true
        },
        city: {
            type: String,
            required: true,
            validate: {
                validator: function(value) {
                    var alphaSpaceRegex = /^[a-zA-Z ]*$/;
                    return alphaSpaceRegex.test(value);
                },
```



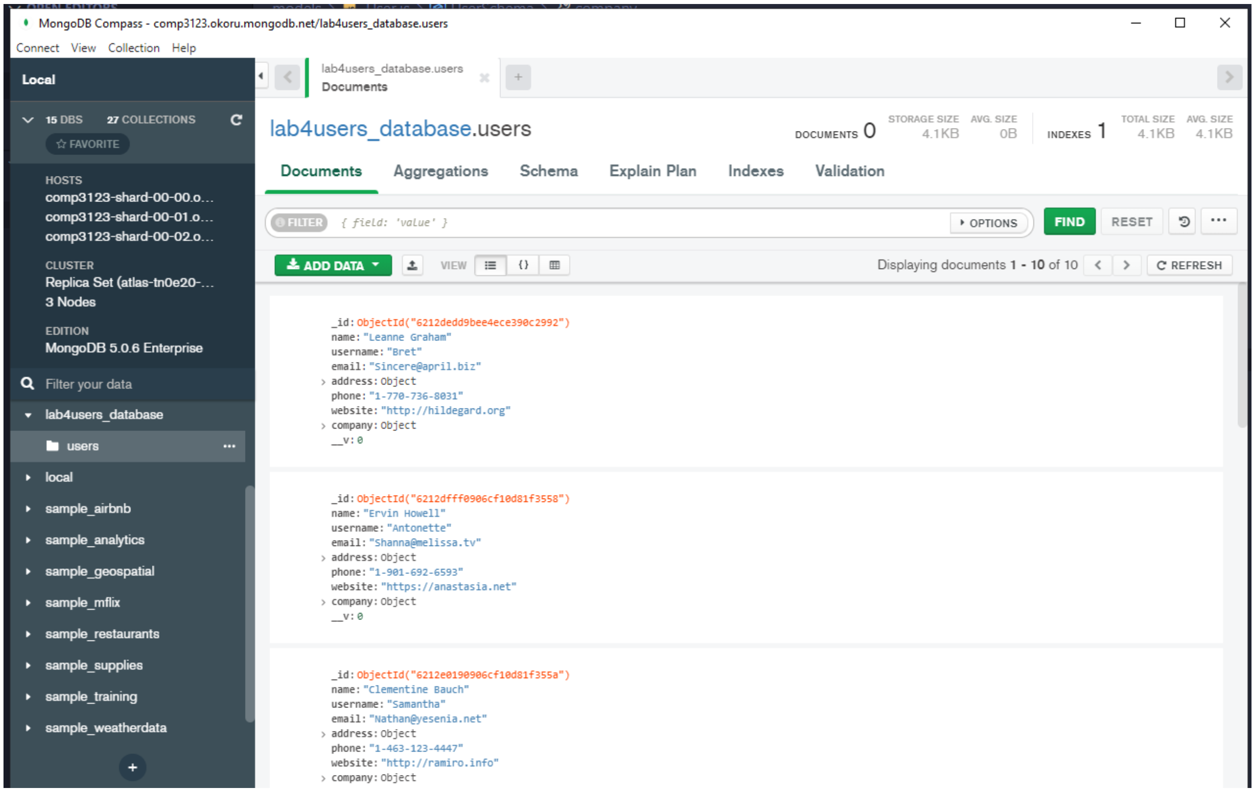
```
        required: true,
        validate: {
            validator: function(value) {
                var webRegex = /^((http|https):\/\/)([A-z]+)\.([A-z]{2,})/;
                return webRegex.test(value);
            },
            message: 'Website must be of valid format'
        }
    },
    company: {
        name: {
            type: String,
            required: true
        },
        catchPhrase: {
            type: String,
            required: true
        },
        bs: {
            type: String,
            required: true
        }
    }
});

const User = mongoose.model('User', UserSchema);
module.exports = User;
```



```
        },
        zipcode: {
            type: String,
            required: true,
            validate: {
                validator: function(value) {
                    var zipRegex = /^\d{5}-\d{4}$/gm;
                    return zipRegex.test(value);
                },
                message: 'Zipcode must be in valid format'
            }
        },
        geo: {
            lat: {
                type: Number,
                required: true
            },
            lng: {
                type: Number,
                required: true
            }
        }
    },
    phone: {
        type: String,
        required: true,
        validate: {
            validator: function(value) {
                var phoneRegex = /^\d{1}-\d{3}-\d{3}-\d{4}$/gm;
                return phoneRegex.test(value);
            },
            message: 'Phone Number must be in valid format'
        }
    },
    website: {
        type: String,
        required: true,
        validate: {
            validator: function(value) {
                var webRegex = /^((http|https):\/\/)([A-z]+)\.([A-z]{2,})/;
```

MongoDB collection and documents:



```
_id: ObjectId("6212dedd9bee4ece390c2992")
name: "Leanne Graham"
username: "Bret"
email: "Sincere@april.biz"
> address: Object
phone: "1-770-736-8031"
website: "http://hildegard.org"
> company: Object
__v: 0

_id: ObjectId("6212dfff0906cf10d81f3558")
name: "Ervin Howell"
username: "Antonette"
email: "Shanna@melissa.tv"
> address: Object
phone: "1-901-692-6593"
website: "https://anastasia.net"
> company: Object
__v: 0

_id: ObjectId("6212e0190906cf10d81f355a")
name: "Clementine Bauch"
username: "Samantha"
email: "Nathan@yesenia.net"
> address: Object
phone: "1-463-123-4447"
website: "http://ramiro.info"
> company: Object
```