

MedioCademy Bootcamp Specialist.

NodeJS Project Shop Online.

#22 POST add Product

Pada materi POST add Product kita ingin mem-fungsikan form "add product" agar bisa menyimpan product baru ke dalam database.

Kita akan update code copas an dari file routes > admin_pages.js:

```
// POST add product
router.post('/add-product', function(req, res) {

    req.checkBody('title', 'Title must have a value').notEmpty();
    req.checkBody('desc', 'Description must have a value').notEmpty();
    req.checkBody('price', 'Price must have a value').isDecimal();
```

Selanjutnya kita juga akan melakukan validasi upload-an image, untuk itu kita akan melakukan custom-validator image pada file **app.js**, di bawah express validator middleware, tepatnya setelah block errorFormater, tambahkan code seperti dibawah ini:

```
customValidators: {
    isImage: function (value, filename) {
        var extension =
(path.extname(filename)).toLowerCase();
        switch (extension) {
            case '.jpg':
                return '.jpg';
            case '.jpeg':
                return '.jpeg';
            case '.png':
                return '.png';
            case '':
                return '.jpg';
            default:
                return false;
        }
    }
}
```

Selanjutnya kita harus validasi juga file upload tidak "undefined", tambahkan variable "imageFile" dibawah router.post(...)...

```
var imageFile = typeof req.files.image !== "undefined" ?
req.files.image.name : "";
```

Kemudian tambahkan require checkBody untuk image:

```
req.checkBody('image', 'You must upload an image!').isImage(imageFile);
```

Lanjut dengan code dibawah ini:

```
        var title = req.body.title;
var slug = title.replace(/\s+/g, '-').toLowerCase();
var desc = req.body.desc;
var price = req.body.price;
var category = req.body.category;

var errors = req.validationErrors();

        if (errors) {
            Category.find(function (err, categories) {
                res.render('admin/add_product', {
                    errors: errors,
                    title: title,
                    desc: desc,
                    categories: categories,
                    price: price
                });
            });
        } else {
            Product.findOne({slug: slug}, function (err, product) {
                if (product) {
                    req.flash('danger', 'Product title
exists, choose another.');
```

```
                Category.find(function (err,
categories) {
                    res.render('admin/add_product', {
                        title: title,
                        desc: desc,
                        categories:
categories,
                        price: price
                    });
                });
            } else {
                var price2 =
parseFloat(price).toFixed(2);
```

```
        var product = new Product({
            title: title,
            slug: slug,
            desc: desc,
            price: price2,
            category: category,
            image: imageFile
        });

        product.save(function (err) {
            if (err)
                return
                console.log(err);

            mkdirp('public/product_images/' + product._id, function (err) {
                return
                console.log(err);

            });

            mkdirp('public/product_images/' + product._id + '/gallery', function (err)
            {
                return
                console.log(err);

            });

            mkdirp('public/product_images/' + product._id + '/gallery/thumbs',
            function (err) {
                return
                console.log(err);

            });

            if (imageFile != "") {
                var productImage =
                req.files.image;
                var path =
                'public/product_images/' + product._id + '/' + imageFile;

                productImage.mv(path, function (err) {
                    return
                    console.log(err);

                });
            }

            req.flash('success',
            'Product added!');
```

```

res.redirect('/admin/products');

    });

    }

    });

```

Kita coba pada browser localhost:3000/admin/products > klik add new button > dan klik submit button, biarkan form kosong.

Maka akan ada 3 buah error message:

- Title must have a value.
- Description must have a value.
- Price must have a value.

Kemudian test untuk upload file selain image yaitu pdf atau yg lainnya:

Maka akan mendapat pesan error berikut nya yaitu "You must upload an image."

Kemudian coba untuk meng-input form product dengan **tanpa** menyertakan image upload > maka akan ditampilkan image **noimage**, selanjutnya nanti kita akan kecilkan ukuran gambarnya.

Coba lagi add product dengan menambahkan image upload, apakah **add product** sukses?

Sekarang kita akan merubah size image noimage.

Buka file views > admin > product.ejs dan update kode dibawah ini dengan memberikan id pada element noimage seperti dibawah ini:

```

<td>
  <% if (product.image == ""){ %>
    
  <% } else { %>
    
  <% } %>
</td>

```

menjadi:

```

<td>
  <% if (product.image == "") { %>
    

```

```
<% } else {%>

<% }%>
</td>
```

Kemudian buka public > CSS > adminstyle.css, tambahkan style sbb:

```
#noimage {
width: 100px;
height: 100px;
}
```

Kemudian refresh localhost:3000/admin/products.

continue next video "Get Edit Product".

Copyright ©MedioCademy.com 2018 Dilarang memperbanyak dan mendistribusikan Materi Training ini tanpa ijin dari MedioSoft Corporation. Pelanggaran akan hak cipta dapat ajukan ke Institusi Hukum yang berlaku.