

➔ Edit database data use (method-override)

Import method override ➔

```
let methodOverride = require('method-override')
```

middleweare ➔

```
app.use(methodOverride('_method'))
```

point ... ➔

किसी mongodb के data को delete करने के लिए delete method का use किया जाता है। लेकिन किसी भी form में 'GET' या 'PUT' method ही होता है। 'DELETE' method बनाने के लिए form के resources action के बाद "`?_method=delete`" का use किया जाता है

```
ex          <form method="POST" action="/delete/<%=y.pageTitle%>?_method=DELETE">

              <button type="submit">Delete</button>

            </form>
```

➔ Example of create, read, edit and delete

Index.js

```
let express = require('express');
let bodyParser = require('body-parser')
let mongoose = require('mongoose')
let session = require('express-session');
let flash = require('connect-flash')
// import methodoverrid
let methodOverride = require('method-override')

let app = express()
app.set('view engine', 'ejs')
app.use(bodyParser.urlencoded({extended:true}))
// middleweare of methodover ride
app.use(methodOverride('_method'))

//montoose conect, schema and model
mongoose.connect('mongodb://localhost:27017/testdb')
let myschema = mongoose.Schema({
  pageTitle :String,
  pageDetails : String
})
let mymodel = mongoose.model('dbtale', myschema)
```

```

//session and flash middle weare
app.use(session({
  secret:'nodejs',
  resave:true,
  saveUninitialized:true
}))
app.use(flash())

// global variable and set message
app.use((req, res, next)=>{
  res.locals.sucess = req.flash('sucess')
  res.locals.err = req.flash('err')
  //Note: yahan 'sucess' & 'err' globally message set kiya jaa raha hai jiska use puer
  application men kahin bhi kiya jaa sakta hai
  next()
})

//route here
app.get('/', (req, res)=>{

  mymodel.find({})
  .then((x)=>{
    res.render('index', {x})
  })
  .catch((y)=>{
    console.log(y)
  })
})

app.get('/add-page', (req, res)=>{
  res.render('add-file')
})

app.post('/add-page', (req, res)=>{
  // here read post for data
  let data = {
    pageTitle : req.body.pagetitle,
    pageDetails : req.body.pagedetails

  }
  mymodel.create(data)
  .then((x)=>{
    req.flash('sucess', 'Your Data has been created on Data base')
    res.redirect('/')
  })
  .catch((y)=>{
    req.flash('err', 'Your Data has been not created')
    res.redirect('/')
  })
})

```

```

}))

//EDIT ROUTE FOR READ OLD DATA
app.get('/edit/:id', (req, res)=>{
  let query = req.params.id;
  mymodel.findOne({pageTitle:query})
  .then((x)=>{
    res.render('edit-file', {x})
    console.log(x)
  })
  .catch((y)=>{
    console.log(y)
  })
})

}))

//PUT ROUTE FOR EDIT DATA
app.put('/edit/:id', (req, res)=>{
  let query = req.params.id;
  mymodel.updateOne({pageTitle:query}, {$set:{
    pageTitle :req.body.pagetitle,
    pageDetails :req.body.pagedetails
  }})
  .then((x)=>{
    req.flash('sucess', 'Your Data has been sucessfully updated')
    res.redirect('/')
    console.log(x)
  })
  .catch((y)=>{
    console.log(y)
  })
})

}))

//DELETE ROUTE FOR DELETE DATA
app.delete('/delete/:id', (req, res)=>{
  let query = req.params.id;
  mymodel.deleteOne({pageTitle:query})
  .then((x)=>{
    req.flash('sucess', 'Your Data has been Deleted')
    res.redirect('/')
  })
  .catch((y)=>{
    console.log(y)
  })
})

}))

app.listen(200, ()=>{
  console.log('200 Port Working')
})

```

add-file.ejs

```
<form method="post" action="/add-page">
  <input type="text" class="form-control" name="pagetitle" placeholder="page Title"><br><br>
  <textarea name="pagedetails" class="form-control" placeholder="page
details.."></textarea><br><br>
  <button>Submit</button>
</form>
```

index.ejs

```
<%if(sucess && sucess !=''){%>
  <div class="alert alert-success" role="alert">
    <%= sucess%>
  </div>
<%}else if(err && err !=''){%>
  <div class="alert alert-danger" role="alert">
    <%= err%>
  </div>
<%}%>

<a href="add-page">Add File</a><br><br>

<table cellpadding='10' cellspacingin='10' border="1">
  <tr>
    <td>Page Title</td>
    <td>Page details</td>
    <td>Edit</td>
    <td>Delete</td>
  </tr>
  <%x.forEach((y)=>{%>
    <tr>
      <td><%=y.pageTitle%></td>
      <td><%=y.pageDetails.slice(0, 20)%></td>
      <td><a href="/edit/<%=y.pageTitle%>">Edit</a></td>
      <td><form method="POST" action="/delete/<%=y.pageTitle%>?_method=DELETE">
        <button type="submit">Delete</button>
      </form></td>
    </tr>
  <%}%>
</table>
<br>
```

Edit-file.ejs

```
<form method="post" action="/edit/<%=x.pageTitle%>?_method=put">
  <input type="text" class="form-control" name="pagetitle" value="<%=x.pageTitle%>"
placeholder="page Title"><br><br>
  <textarea name="pagedetails" class="form-control" value="<%=x.pageDetails%>" placeholder="page
details.."><%=x.pageDetails%></textarea><br><br>
  <button>Submit</button>
</form>
```