

WEB APPLICATION MINOR 2

exp_index.js:

```
const express = require('express');
const bodyParser = require('body-parser');
const ejs = require('ejs');
const mongoose = require('mongoose');
const bcrypt = require('bcrypt');

/* Restapi */
const jobsApi = require('./routes/jobsRoute')

const app = express();
const port = 3000;

app.use('/jobs',jobsApi)
```

jobsRoute.js:

```
const express = require('express')
const router = express.Router()
const mongoose = require('mongoose')
const bodyParser = require('body-parser');
router.use(bodyParser.urlencoded({extended:true}));

router.get('/',async(req,res)=>{
```

```
    let data = await JobDetail.find()  
    res.json(data)  
  })
```

```
const jobDetailSchema = new mongoose.Schema({  
  jobtitle:{  
    type:String  
  
  },  
  jobdescription:{  
    type:String  
  },  
  salary:{  
    type:Number  
  },  
  experience:{  
    type:Number  
  
  },  
  location:{  
    type:String  
  }  
})
```

```
const JobDetail = mongoose.model('JobDetail',jobDetailSchema)  
router.post('/',async (req,res)=>{
```

```

let jobdata = {
  jobtitle: req.body.jobtitle,
  jobdescription : req.body.jobdescription,
  salary: req.body.salary,
  experience : req.body.experience,
  location: req.body.location
}
const newjob = new JobDetail(jobdata)
newjob.save().then(response=>{
  res.json(response)
}).catch(err=>{
  if(err){
    console.log("error")
  }
})
})

```

```

router.get('/:id',async(req,res)=>{
  let id = req.params.id
  let data = await JobDetail.findById(id)
  res.json(data)
})

```

```

router.put('/:id',async(req,res)=>{
  let id = req.params.id
  let data = {

```

```

    jobtitle: req.body.jobtitle,
    jobdescription : req.body.jobdescription,
    salary: req.body.salary,
    experience : req.body.experience,
    location: req.body.location
  }
  await JobDetail.findByIdAndUpdate(id,data,(err,docs)=>{
    if(!err){
      res.json("Updated Successfully")
    }else{
      res.json('Failed to update')
    }
  })
})

```

```

router.delete('/:id',async(req,res)=>{
  var id = req.params.id
  await JobDetail.findByIdAndDelete(id,(err,docs)=>{
    res.json(docs)
    if(!err){
      if(id===JSON.stringify(docs)[ '_id']){
        res.json("deleted successfully")

      }
      else if(JSON.stringify(docs)=== 'null'){

```

```

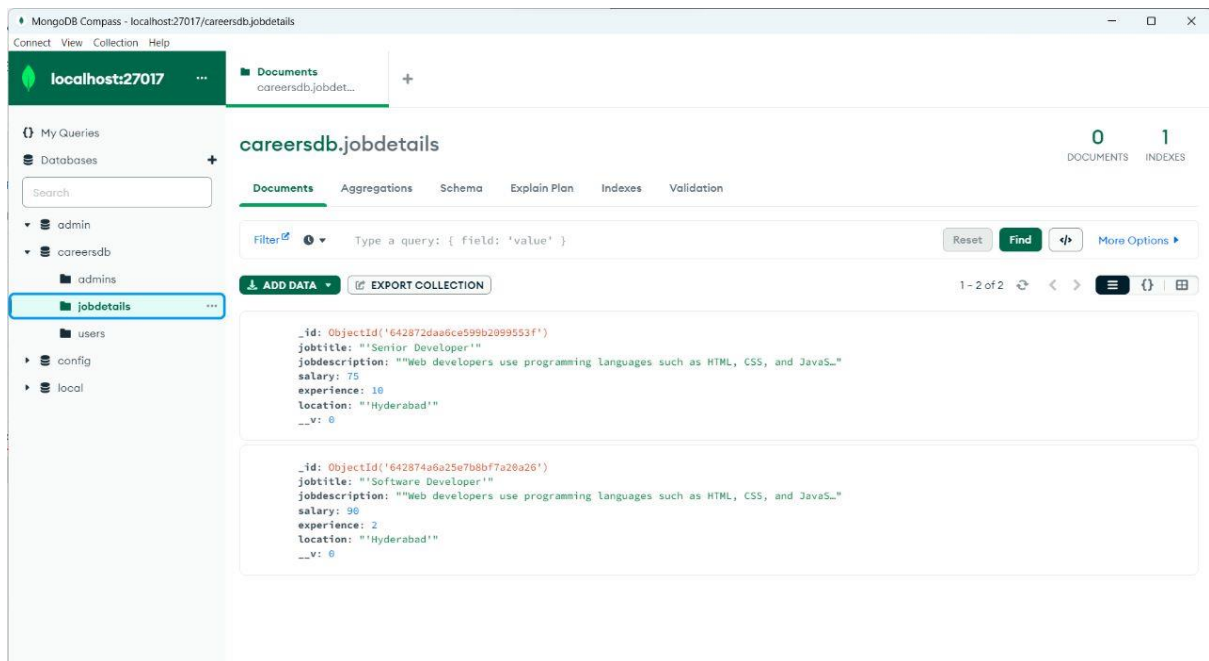
        res.json('NO data to delete')
    }
}
else{
    res.json(err)
}
})
})

module.exports = router

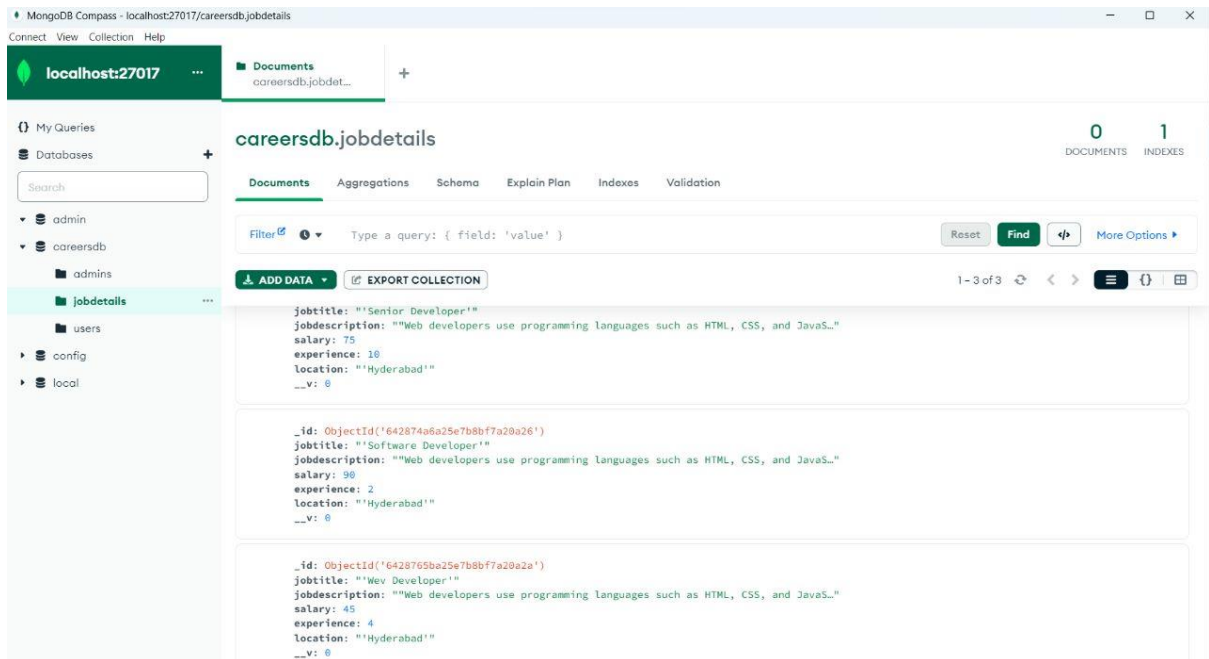
```

SCREENSHOTS:

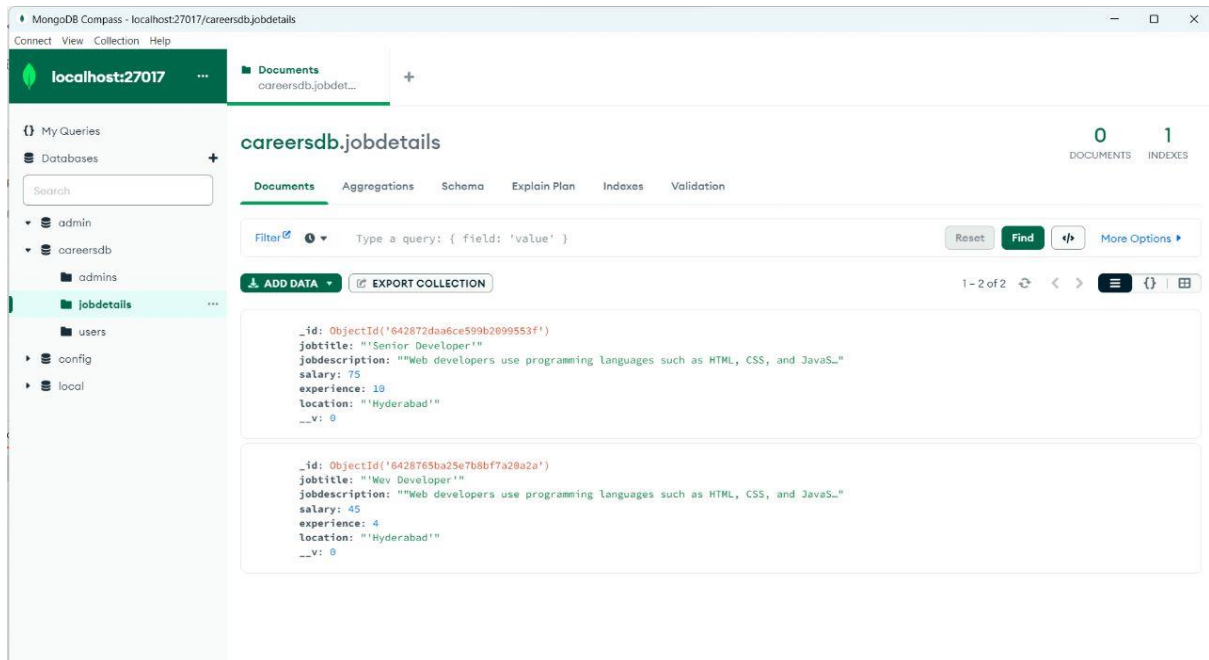
Before post operaton:



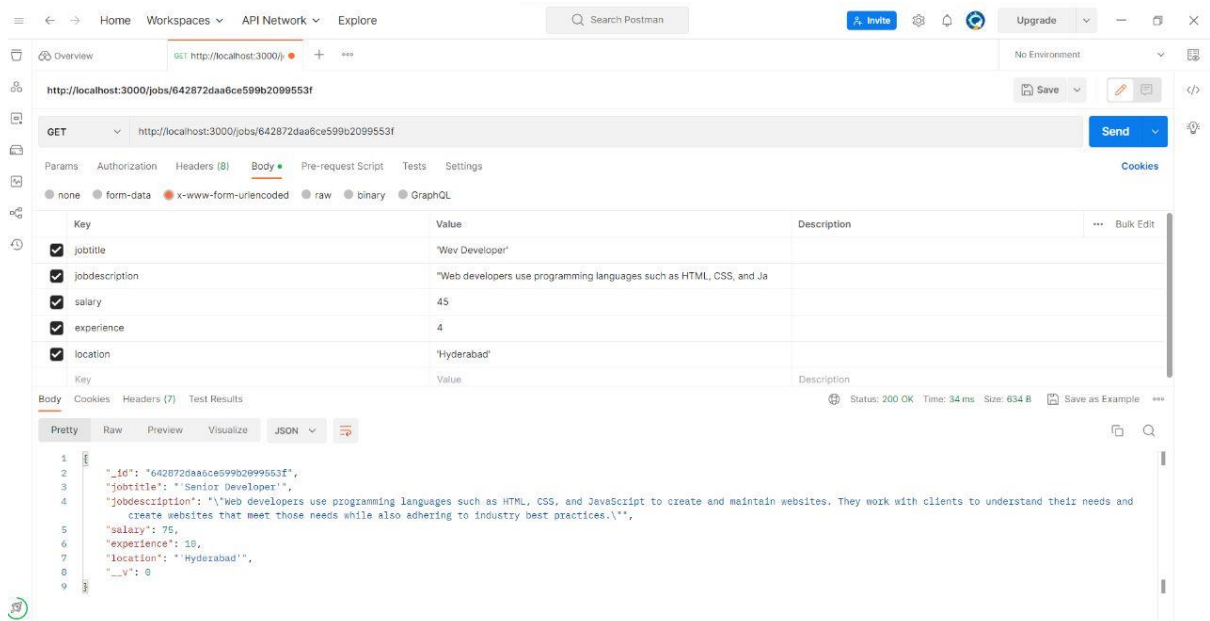
After post operation:



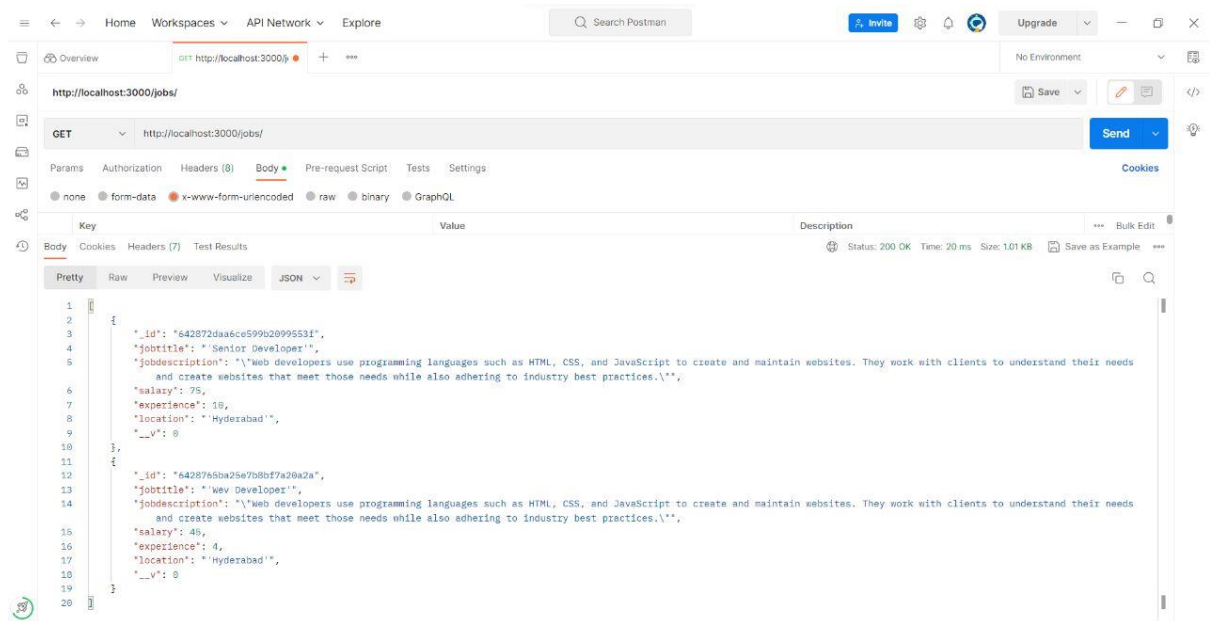
After deleting software developer job:



Getting by id (senior developer job):



Getting all jobs in database:



SUBMITTED BY

Nandam Sai Saketh

420212