**==> npm init**

**==> npm install express --save**

**==>npm install body-parser --save**

**==> npm install mongoose --save**

**Config.js**

**module.exports={**

**port:3000**

**}**

**App.js**

**const express=require("express");**

**const app=express();**

**var route = express.Router();*//Initialize route object***

**var config=require('./config');**

**//need body parser to take get and post parameters**

**const bodyparser=require("body-parser");**

**app.use(bodyparser.urlencoded({extended:true}));**

**app.use(bodyparser.json());**

**//to connect mongodb**

**const mongoose=require("mongoose");**

***//mongodb connection***

**mongoose.connect('mongodb://localhost/databaseName', { useNewUrlParser: true },function(err){**

**if(err){**

**console.log("The database error is: ",err);**

**}else{**

**console.log("The database connected successfully!");**

**}**

**})**

**route.post('/login',function(req,res){**

**var email = req.body.email;**

**var pass = req.body.pass;**

**console.log(email,",",pass)**

**if(email == 'kumarrajeev.191@gmail.com' && pass == '12345'){**

**res.json({**

**success : true,**

**message : 'correct login',**

**})**

**}**

**else{**

**res.json({**

**success : false,**

**message : 'Incorrect login'**

**})**

**}**

**})**

**route.get('/users',function(req,res){**

**Console.log("inside user get call");**

**})**

**route.get('/users/:id',function(req,res){**

**Const userId=body.params.id;**

**Console.log("inside user get call by userid: "+userId);**

**})**

**route.put('/users/:id',function(req,res){**

**Const userId=body.params.id;**

**Console.log("inside user update call by userid: "+userId);**

**})**

**route.delete('/users/:id',function(req,res){**

**Const userId=body.params.id;**

**Console.log("inside user delete call by userid: "+userId);**

**})**

**app.use('/api',router);**

**app.listen(config.port,function(){**

**console.log('The server is listening port: ',config.port);**

**})**

**//using mongo db**

**==> first need to create model for each table which we are going to use**

**==> create models folde inside root folder**

**==> suppose we are going to use userdetails table, then we will create users.model.js inside models folder**

**User.js**

**const mongoose=require("mongoose");**

**const schema=mongoose.Schema;**

**const UserModel=new schema({**

**first\_name:{type: String},**

**last\_name:{type: String},**

**email:{type: String,unique:true,lowercase:true},**

**pic:{type: String},**

**dob:{type: Date},**

**mobile:{type: String},**

**});**

**module.exports=mongoose.model('User',UserModel);**

**==> then we will call this user.js models file into app.js file**

**var USER=require('./models/User.model');**

**router.get('/users',function(req,res){**

**USER.find({},function(err,users){**

**if(err){**

**res.json({**

**status:false,**

**message:err**

**})**

**}else{**

**res.json({**

**status:true,**

**records:users**

**})**

**}**

**})**

**})**

**router.get('/users/:id',function(req,res){**

**USER.findById(req.params.id, function (err, users) {**

**if(err){**

**res.json({**

**status:false,**

**message:err**

**})**

**}else{**

**res.json({**

**status:true,**

**records:users**

**})**

**}**

**})**

**})**

**router.post('/users',function(req,res){**

**var usr=new USER();**

**usr.first\_name=req.body.first\_name;**

**usr.last\_name=req.body.last\_name;**

**usr.email=req.body.email;**

**usr.pic=req.body.pic;**

**usr.dob=req.body.dob;**

**usr.mobile=req.body.mobile;**

**usr.save();**

**res.json({**

**status:true,**

**message:'The record inserted successfully!'**

**})**

**})**

**router.put('/users/:id',function(req,res){**

**USER.findByIdAndUpdate(req.params.id, {$set: req.body}, function (err, users) {**

**if(err){**

**res.json({**

**status:false,**

**message:err**

**})**

**}else{**

**res.json({**

**status:true,**

**records:users**

**})**

**}**

**});**

**})**

**router.delete('/users/:id',function(req,res){**

**USER.findByIdAndRemove(req.params.id, function (err) {**

**if(err){**

**res.json({**

**status:false,**

**message:err**

**})**

**}else{**

**res.json({**

**status:true,**

**message:'Deleted successfully!'**

**})**

**}**

**})**

**})**