

GAURAV THANDANI PWSKILL

GENERATE JSONWEBTOKEN (REGISTER/LOGIN/VIEW-DETAILS)

Model:

Models/userModel.js

```
var mongoose=require('mongoose');
var UserSchema=new mongoose.Schema({
  name:String,
  email:String,
  password:String
},{timestamps:true})

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

Controller/UserController.js

```
const userModel = require("../Models/userModel")
const jwt=require('jsonwebtoken');

const handleUserSignUp=(req,res)=>{
  try{
    userModel.create(req.body).then(() =>{
      res.json({"Message":"Created Successfully"}).status(201);
    }).catch((err) =>{
      res.json({"Message":"This is wrong", err:err}).status(500);
    })
  }
  catch(err){
    res.json({"Message":"This is wrong", err:err}).status(500);
  }
}
```

```
}  
}
```

```
const handleUserLogin=(req, res)=>{  
  let {email} = req.body; // destruction of object  
  try{  
    userModel.find({email:email}).then((response)=>{  
  
      if(response.length>=1){  
        jwt.sign(req.body,process.env.SECRET_KEY,(err, token)=>{  
          if(err){  
            res.json({"Message":"This is wrong", err:err}).status(500);  
          }  
          else{  
            res.json({  
              Message:"Login Successfull",  
              data:req.body,  
              token:token,  
            })  
          }  
        })  
      }  
    })  
  }  
  catch(err){  
    res.json({ Message: "This is wrong", err: err }).status(500);  
  
  }  
}
```

```
const getDetails=(req, res)=>{  
}
```

```
module.exports={handleUserSignUp, handleUserLogin, getDetails}
```

Middleware/verifyMiddleware.js

```
const jwt=require('jsonwebtoken');  
const verify=(req,res, next)=>{  
  const tokencheck=req.headers['authorization'];  
  const extractedToken=tokencheck.split(' ')[1]; //["Bearer, <token>"]  
  if(extractedToken){  
    jwt.verify(extractedToken,process.env.SECRET_KEY,(err, decode)=>{  
      if(err){  
        res.json({ Message: "This is wrong", err: err }).status(500);  
      }  
      else{  
        req.email=decode.email  
        next();  
      }  
    })  
  }  
}  
module.exports=verify;
```

Routes/UserRoute.js

```
var express= require('express');  
const { handleUserSignUp, handleUserLogin, getDetails } =  
require('../Controller/UserController');  
const verify = require('../Middleware/verifyMiddleware');
```

```
var router=express.Router();
router.post('/signUp', handleUserSignUp)
router.post('/login', handleUserLogin)
router.get('/getDetails',verify,getDetails)
module.exports=router;
```

server.js

```
var express=require('express');
var app=express();
var dotenv=require('dotenv');
const router = require('./Routes/UserRoute');
var bodyParser=require('body-parser');
var mongoose=require('mongoose');
dotenv.config();
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({extended:true}));
mongoose.connect('mongodb+srv://garvthadani:123Gaurav@cluster0.sa0jbba.mongodb.net/?re
tryWrites=true&w=majority&appName=Cluster0', {
  // string options
  useUnifiedTopology:true,
  useNewUrlParser:true
}).then(())=>{
  console.log("connected to db")
})

app.use('/api/v1/user',router); // router middleware

app.listen(process.env.PORT,()=>{
  console.log(`listening to port ${process.env.PORT}`)
```

```
})  
// http://localhost:portnumber/api/v1/user/signUp  
// http://localhost:portnumber/api/v1/student/signUp  
// packages :- npm i body-parser nodemon jsonwebtoken express mongoose  
/* ENV  
PORT=5002  
SECRET_KEY="123Gaurav"  
*/
```

package.json

```
{  
  "name": "decodenode",  
  "version": "1.0.0",  
  "description": "",  
  "main": "index.js",  
  "scripts": {  
    "test": "echo \"Error: no test specified\" && exit 1",  
    "start": "nodemon server.js"  
  },  
  "keywords": [],  
  "author": "",  
  "license": "ISC",  
  "dependencies": {  
    "body-parser": "^1.20.2",  
    "dotenv": "^16.4.5",  
    "express": "^4.18.3",  
    "jsonwebtoken": "^9.0.2",  
    "mongoose": "^8.2.1",  
    "nodemon": "^3.1.0"  
  }  
}
```

}