

```
//Todo Controller
```

```
import Todo from "../model/todoModel.js";
```

```
export const CreateTodo= async(req,res)=>{  
  const {text}=req.body;  
  const userId=req.userId;  
  try  
  {  
    const newTodo=new Todo({text,userId});  
    await newTodo.save();  
    res.status(200).json({message:"New todo added", todo:newTodo});  
  }  
  catch (error) {  
    console.error("Create todo error",error);  
    res.status(500).json({message:"Internal server error"});  
  }  
}
```

```
export const GetAllTodos=async (req,res)=>{  
  const userId=req.userId;  
  try {  
    const todos=await Todo.find({userId}).populate("userId","firstName lastName email");  
    res.status(200).json({message:"Todo list fetched",todoInfo:todos});  
    console.log("todolist",todos);  
  } catch (error) {  
    console.error("Get todo error",error);  
  }  
}
```

```
    res.status(500).json({message:"Internal server error"});  
  }  
}
```

```
export const UpdateTodo=async(req,res)=>{  
  const {Id}=req.params;  
  const text=req.body;  
  const userId=req.userId;  
  
  try  
  {  
    const updatedTodo=await Todo.findByIdAndUpdate(Id,{text},{new:true})  ;  
    if(!updatedTodo)  
    {  
      return res.status(400).json({message:"Todo Not Found or Unauthorized"});  
    }  
    res.status(200).json({message:"Todo Updated Successfully",todo:updatedTodo});  
  }  
  catch (error) {  
    console.error("Update todo error",error);  
    res.status(500).json({message:"Internal server error"});  
  }  
}
```

```
export const DeleteTodo=async (req,res)=>{  
  const {Id}=req.params;
```

```

const userId=req.userId;

try {

    const deletedTodo=await Todo.findByIdAndDelete({_id:Id,userId});

    if(!deletedTodo)

    {

        return res.status(400).json({message:"Todo Not Found or unauthorized"});

    }

    res.status(200).json({message:"Todo Deleted Successfully"});

} catch (error) {

    console.error("Delete todo error",error);

    res.status(500).json({message:"Internal server error"});

}

}

```

//User Controller

```

import express from "express";

import User from "../model/userModel.js";

import bcrypt from "bcrypt";

import jwt from "jsonwebtoken";

export const SignUp=async (req,res)=>{

    const {firstName,lastName,email,password}=req.body;

    console.log(".....response.....",req.body);

    try {

```

```
const existingUser=await User.findOne({email});

if(existingUser)

{

    return res.status(400).json({message:"User already exists"});

}

const hashPassword=await bcrypt.hash(password,10);


const newUser=new User({

    firstName,

    lastName,

    email,

    password:hashPassword

});

await newUser.save();

res.status(201).json({message:"User Registered Successfully"});

}

catch (error) {

    console.error("sign-up error",error);

    res.status(500).json({message:"Internal Server Error"});

}

}

export const SignIn= async(req,res)=>{

    const {email,password}=req.body;

    try

    {

        const user=await User.findOne({email});
```

```

if(!user)
{
    return res.status(400).json({message:"User Not Found"});
}

const isPasswordValid=await bcrypt.compare(password,user.password);
if(!isPasswordValid)
{
    return res.status(400).json({message:"Invalid email or password"});
}

const token=jwt.sign({userId:user._id},"SECRET_KEY",{expiresIn:"1h"});
res.status(200).json({message:"Login Successful",token});
}

catch (error) {
    console.error("sign-in error",error);
    res.status(400).json({message:"Internal server error"});
}
}

```

//Auth Middleware

```

import jwt from "jsonwebtoken";

const authMiddleware=(req,res,next)=>{

    const token= req.header("Authorization)?.split(" ")[1];

    if(!token)
    {

```

```
    return res.status(401).json({message:"No Token Found,Authorization denied"});
  }
  try
  {
    const decoded=jwt.verify(token,"SECRET_KEY");
    req.userId=decoded.userId;
    next();
  }
  catch (error) {
    console.error("Token verification error",error);
    res.status(401).json({message:"Invalid Token"});
  }
}
export default authMiddleware;
```

//Server.js

```
import express from "express";
import cors from "cors"
// import dotenv from "dotenv";
import con from "./db/connection.js";
import userRouter from "./routes/user.route.js";
import todoRouter from "./routes/todo.route.js";
const port=3000;
```

```
const app=express();
```

```
app.use(cors());
```

```
app.use(express.urlencoded({extended:true}));
```

```
app.use(express.json());
```

```
app.get("/",(req,res)=>{
```

```
    res.send(`<h3>Welcome to To Do App</h3>`)
```

```
});
```

```
app.use("/api/v1/user",userRouter);
```

```
app.use("/api/v1/todo",todoRouter);
```

```
app.listen(port,()=>{
```

```
    console.log(`Server has started at localhost ${port}`);
```

```
});
```