BLOGGING WEBSITE

1. CONFIGURE CODE :

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

//we need to feed this some where else code will have error
require("dotenv").config();
// function to establish connection between application and database
const dbConnect=()=>{
    mongoose.connect(process.env.DATABASE_URL,{
        useNewUrlParser:true,
         useUnifiedTopology:true, // nolonger needein mongodb
    })
    .then(()=>console.log("DB ka Connection is Successful"))
    .catch((error)=>{
        console.log("Issue in DB Connection");
        console.error(error.message);
        process.exit(1);
    });
}
module.exports=dbConnect;
```

2.CONTROLLERS CODE :

```
// import the model
//POST REQUEST
const blog=require("../models/blog");

//define route handeler
exports.createblog=async(req,res)=>{
    try{
        //extract title and content from request body
        const{categories,title,content,author}=req.body;

        //create a new blog and insert in DB
        const response=await blog.create({categories,title,content,author});

        //send a json response with a success flag
        res.status(200).json(
            {
                success:true,
                data:response,
                message:'Entry Created Successfully'
            }
        );
    }
    catch(err){
                console.error(err);
                console.log(err);
                res.status(500)
                .json({
                    success:false,
                    data:"internal server error",
                    message:err.message,
                })
    }
}
```

```
//delete REQUEST BYY  ID

const blog=require("../models/blog");

//define route handeler
exports.deleteblog=async(req,res)=>{
    try{
        const {id}=req.params;
        await blog.findByIdAndDelete(id);
        res.json({
            success:true,
            message:"Blog DELETED",
        });
    }
    catch(err){
        console.error(err);
        res.status(500).json({
            success:false,
            error:err.message,
            message:'Server Error',
        });
}
};
```

```
// import the model
//GET REQUEST
const Blog = require("../models/blog");

// define route handler
exports.getblog = async (req, res) => {
    try {
        // fetch all blog items from the database
        const blogs = await Blog.find({});

        // response
        res.status(200).json({
            success: true,
            data: blogs,
            message: "Entire blog is fetched",
        });
    } catch (err) {
        console.error(err);
        res.status(500).json({
            success: false,
            error: err.message,
            message: 'Server Error',
        });
    }
}

exports.getblogById = async (req, res) => {
    try {
        // extract blog item based on id
        const id = req.params.id;
        const blogItem = await Blog.findById({ _id: id })

        // data for given id not found
        if (!blogItem) {
            return res.status(404).json({
                success: false,
                message: "NO DATA FOUND FOR GIVEN ID",
            })
        }
        // data for given id found
        res.status(200).json({
            success: true,
            data: blogItem,
            message: `Blog ${id} data successfully fetched`,
        })
    } catch (err) {
        console.error(err);
        res.status(500).json({
            success: false,
            error: err.message,
            message: 'Server Error',
        });
    }
}
```

```
// import the model
//PUT  REQUEST (USING ID)
const Blog=require("../models/blog");
```

```
//define route handeler
exports.updateblog=async(req,res)=>{
    try{
        const{id}=req.params;
        const{categories,title,content,author}=req.body;

        const blog =await   Blog.findByIdAndUpdate(
            {_id:id},
            {categories,title,content,author,updatedAt:Date.now()},
        )
        res.status(200).json({
            success:true,
            data:blog,
            message:"UPDATED SUCCESFULLY",
        });
    }
    catch(err){
        console.error(err);
        res.status(500)
        .json({
            success:false,
            error:err.message,
            message:'Server Error',
        });
}
}
```

3. MODELS CODE:

```
const mongoose = require('mongoose');

// Define a schema for the blog post
const blogSchema = new mongoose.Schema({
    categories: {
        type: String,
        required: true,
      },
  title: {
    type: String,
    required: true,
  },
  content: {
    type: String,
    required: true,
  },
  author: {
    type: String,
    required: true,
  },
  createdAt: {
    type: Date,
    default: Date.now(),
  },
  updatedAt:{
    type:String,
    required:true,
    default:Date.now(),
  }
});

// Export the model for use in other files with name of blog
module.exports = mongoose.model("Blog",blogSchema);
```

4. ROUTES CODE:

```
const express = require("express");
const router =express.Router();

//import controller
const {createblog}=require("../controllers/createblog");
const {getblog,getblogById}=require("../controllers/getblog");
const {updateblog}=require("../controllers/updateblog");
const {deleteblog}=require("../controllers/deleteblog");

//define API routes
router.post("/createblog",createblog);
router.get("/getblog",getblog);
router.get("/getblog/:id",getblogById);
router.put("/updateblog/:id",updateblog);
router.delete("/deleteblog/:id",deleteblog);
module.exports=router;
```

5. INDEX.JS CODE:

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

//load config from env file
require("dotenv").config();
const PORT=process.env.PORT || 4000;

// middleware to parse json request body
app.use(express.json());
```

```
//import routes for BLOG API
const blogRoutes=require("./routes/blogs");

//mount the BLOG ASPI routes
app.use("/",blogRoutes);

//start server
app.listen(PORT,()=>{
    console.log(`Server started successfully at ${PORT}`);
})

//connect to the database
const dbConnect=require("./config/database");
dbConnect();

// default Route
app.get("/",(req,res)=>{
    res.send(`<h1>This is HomePage</h1>`)
})
```