Name: KHUSHI POLSHETTIWAR
Class: TE IT Div: 2
Roll No.: I3236
Assignment: 3B

Problem Statement: Create four API using Node.JS, ExpressJS and MongoDB for CURD Operations on assignment 2.C

Code:

```javascript
const express = require("express");
const bodyParser = require("body-parser");
const ejs = require("ejs");
const mongoose = require("mongoose");

const app = express();   //Creating new app instance

app.set('view engine', 'ejs');

app.use(bodyParser.urlencoded({extended: true}));
app.use(express.static("public"));

mongoose.set('strictQuery',false);
mongoose.connect("mongodb://127.0.0.1:27017/wikiDB");

const articleSchema = {
  title:String,
  content:String
}

const Article = mongoose.model("Article",articleSchema);

///////////////////////////////// REQUESTS TARGETTING ALL ARTICLES
/////////////////////////////////////

app.route("/articles")
```

```javascript
.get(function(req,res){
  Article.find(function(err,foundArticles){
    if(!err){
        res.send(foundArticles);
    }else{
        res.send(err);
    }
  });
})



.post(function(req,res){

  const newArticle = new Article({
    title:req.body.title,
    content:req.body.content
  });
  newArticle.save(function(err){
    if(!err){
        res.send("Successfully added a new Article");
    }else{
        res.send(err);
    }
  });
})
```

```javascript
    if(!err){
      res.send("Successfully added a new Article");
    }else{
      res.send(err);
    }
  });
})

.delete(function(req,res){
  Article.deleteMany(function(err){
    if(!err){
      res.send("Successfully deleted all Articles");
    }else{
      res.send(err);
    }
  });
});


//////////////////////////// REQUESTS TARGETTING A SPECIFIC ARTICLE
////////////////////////////////////

app.route("/articles/:articleTitle")
```

```
.get(function(req,res){
Article.findOne({title:req.params.articleTitle},
function(err,foundArticle){
    if(foundArticle){
        res.send(foundArticle)
    }else{
        res.send("No Articles matching that title found");
    }
});
})

.put(function(req,res){
Article.findOneAndUpdate(
    {title: req.params.articleTitle},
    {title: req.body.title, content: req.body.content},
    {overwrite:true},
    function(err){
        if(!err){
            res.send("Successfully updated  article")
        }else{
            res.send(err)
        }
    }
);
})

.patch(function(req,res){
Article.findOneAndUpdate(
    {title: req.params.articleTitle},
    {$set:req.body},
    function(err){
        if(!err){
            res.send("Successfully updated")
        }else{
            res.send(err)
        }
    }
);
});
```
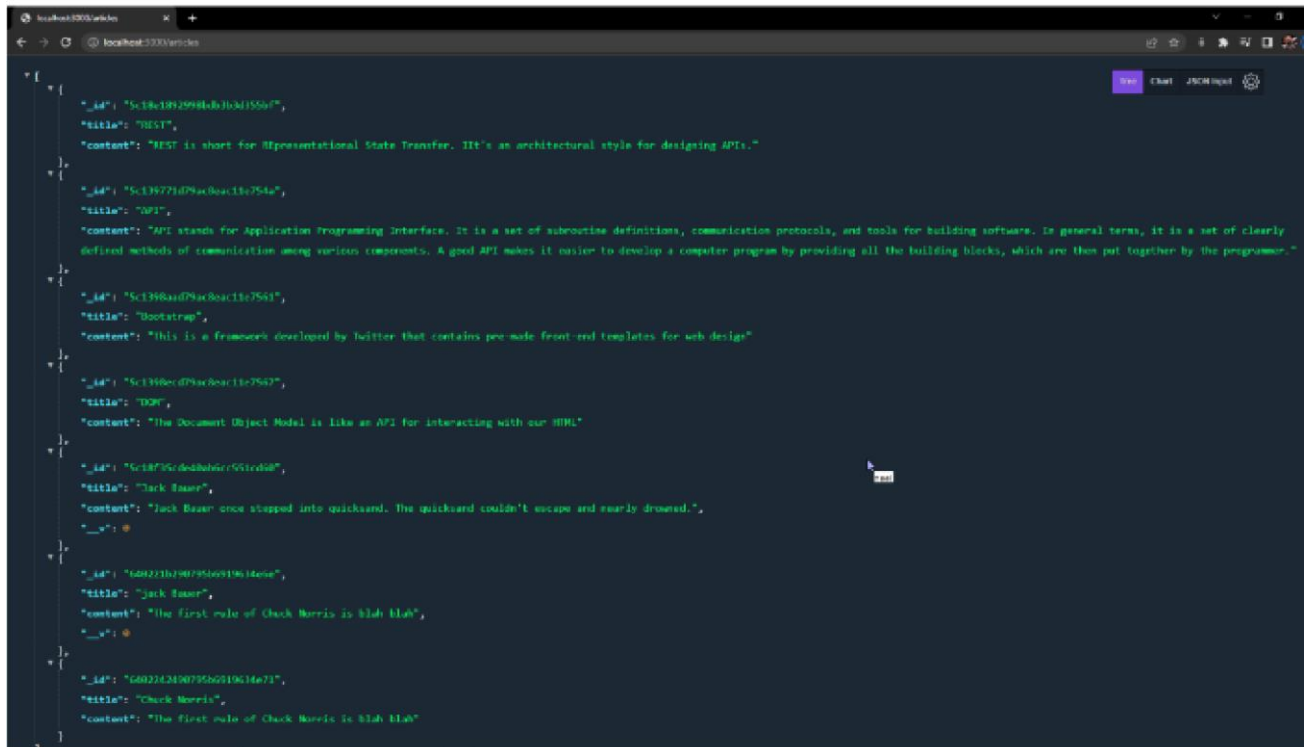
```
app.listen(3000,function(req,res){
  console.log("Server started at port 3000");
});
```

Output: