Index.js

```
const express = require('express');
const mongoose = require('mongoose');
const app=express();
app.use(express.json);
app.listen(3000,()=>{
    console.log("OK");
})

require('dotenv').config();

const mongoString = process.env.DATABASE_URL
mongoose.connect(mongoString);
const database = mongoose.connection
database.on('error', (error) => {
    console.log(error)
})

database.once('connected', () => {
    console.log('Database Connected');
})
const routes = require('./routes/routes');

app.use('/api', routes)
```

route.js

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

const router = express.Router()

module.exports = router;
//Post Method
router.post('/post', (req, res) => {
    res.send('Post API')
})

//Get all Method
router.get('/getAll', (req, res) => {
    res.send('Get All API')
})

//Get by ID Method
router.get('/getOne/:id', (req, res) => {
    res.send('Get by ID API')
})
```

```
//Update by ID Method
router.patch('/update/:id', (req, res) => {
    res.send('Update by ID API')
})

//Delete by ID Method
router.delete('/delete/:id', (req, res) => {
    res.send('Delete by ID API')
})
const Model = require('../models/model');
//Post Method
router.post('/post', (req, res) => {
    const data = new Model({
        name: req.body.name,
        age: req.body.age
    })

    try {
        const dataToSave = data.save();
        res.status(200).json(dataToSave)
    }
    catch (error) {
        res.status(400).json({message: error.message})
    }
})
```

Model.js

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

const dataSchema = new mongoose.Schema({
    name: {
        required: true,
        type: String
    },
    age: {
        required: true,
        type: Number
    }
})

module.exports = mongoose.model('Data', dataSchema)
```

app.js (API fetching for demo purpose)

```
import React,{useState,useEffect} from "react";
import List from "./List";
import axios from "axios";
```

```
function App() {
  var [card,Setcard]=useState([]);
  const
token="eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiIsImtpZCI6IjI4YTMxOGY3LTAwMDAtYTFlYi0
3ZmExLTJjNzQzM2M2Y2NhNSJ9.eyJpc3MiOiJzdXBlcmNlbGwiLCJhdWQiOiJzdXBlcmNlbGw6Z2Ft
ZWFwaSIsImp0aSI6IjZmMDhhM2Y4LTg1NDgtNGM4Ny04MjhmLTI5YWZkMzcwNGViMSIsImlhdCI6MT
Y5NjY2NzIyNCwic3ViIjoiZGV2ZWxvcGVyLzIzZWI3ZWM3LTk3NjYtOTMzNy1lNDRhLTVlODcwZDk2
ODY5MyIsInNjb3BlcyI6WyJyb3lhbGUiXSwibGltaXRzIjpbeyJ0aWVyIjoiZGV2ZWxvcGVyL3NpbH
ZlciIsInR5cGUiOiJ0aHJvdHRsaW5nIn0seyJjaWRycyI6WyI0OS4yMDQuMTE1LjIzNCJdLCJ0eXBl
IjoiY2xpZW50In1dfQ._lyLdjqezx2tGH9j5tvH0RWx5KUTGL7DEM8KWMSvx0f4YboMEXxB-
uGbIRFK1qy_jNELhSPLAWTN710DgNoPtw";
  var cors=require('cors');
  const [currenturl,setcurrenturl]=
useState("https://proxy.royaleapi.dev/v1/cards?limit=20")
  const [load,setload]=useState(true)

  //var express=require('express');
  //var app=express();
  //App.use(cors());
  const cfg={
    headers:{
      "Access-Control-Allow-Origin": "*",
    "Access-Control-Allow-Methods": "GET,PUT,POST,DELETE,PATCH,OPTIONS",

    "Access-Control-Allow-Headers":"Content-Type,X-Auth-
Token,Origin,Authorization",
      "Authorization":"Bearer
${eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiIsImtpZCI6IjI4YTMxOGY3LTAwMDAtYTFlYi03ZmEx
LTJjNzQzM2M2Y2NhNSJ9.eyJpc3MiOiJzdXBlcmNlbGwiLCJhdWQiOiJzdXBlcmNlbGw6Z2FtZWFwa
SIsImp0aSI6IjhiYTE3NGI4LTZhOWQtNDViMi1iZjkyLTVlMWNiNDM4NWM2ZiIsImlhdCI6MTY5Njc
wOTA4OSwic3ViIjoiZGV2ZWxvcGVyLzIzZWI3ZWM3LTk3NjYtOTMzNy1lNDRhLTVlODcwZDk2ODY5M
yIsInNjb3BlcyI6WyJyb3lhbGUiXSwibGltaXRzIjpbeyJ0aWVyIjoiZGV2ZWxvcGVyL3NpbHZlciI
sInR5cGUiOiJ0aHJvdHRsaW5nIn0seyJjaWRycyI6WyI0NS43OS4yMTguNzkiXSwidHlwZSI6ImNsa
WVudCJ9XX0.2EXhoFupm88uj_ClTD3hzNnDYWhGTMog9_0y9Q5pnlHd4O2qkyLyNaUF2QvbBZdHP5i
J0PNFZvBip8mYFItOvg"

    }

  }
  useEffect(()=>{//Fetch once //[] everytime cururl change fetch
    setload(true)
    let c1
    axios.get(currenturl,cfg,
      {
        cancelToken: new axios.CancelToken(c =>c1=c)
      }
      ).then(
   response =>{
    setload(false)
    Setcard(response.data.items.map(p=>p.iconUrls.medium))
    console.log(card)
   }

   ).catch(err=>console.log(err))
   return ()=> c1()
 },[currenturl])
 if(load) return  "Loading..."

 return (
  <List card={card}></List>
 );
}

export default App;
```

index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';

import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
 <React.StrictMode>
  <App />
 </React.StrictMode>
);

// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
```

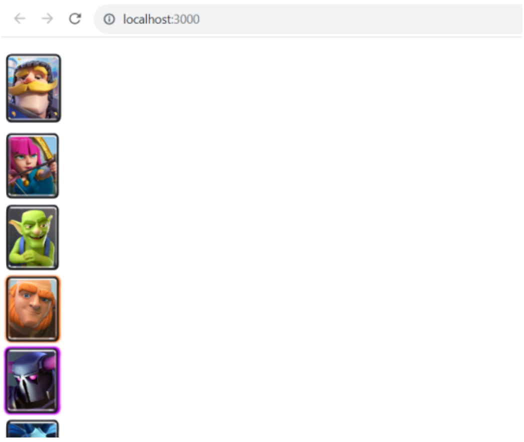
List.js

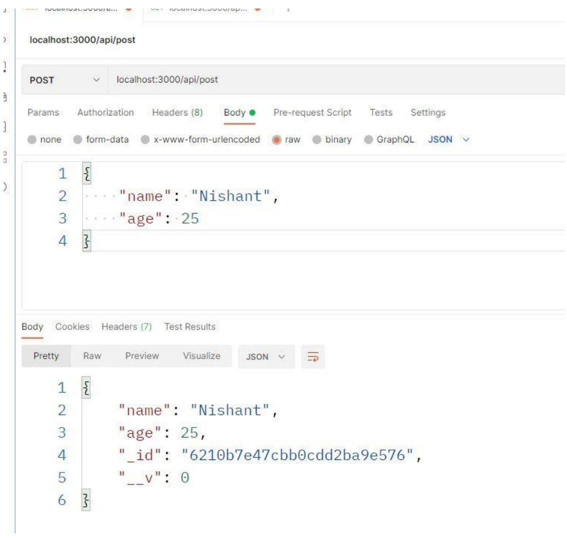
```
import React from 'react'

export default function List({card}) {

  return (
    <div>
  {card.map(p=>(
    <div key={p}>
      <img src={p}></img>
    </div>
    ))}
    </div>
  )
}
```

Output




localhost:3000/api/post

POST localhost:3000/api/post

Params
Authorization
Headers (8)
Body ●
Pre-request Script
Tests
Settings
○ none ○ form-data ○ x-www-form-urlencoded ● raw ○ binary ○ GraphQL JSON

```
{
    "name": "Nishant",
    "age": 25
}
```

```
{
    "name": "Nishant",
    "age": 25,
    "_id": "6210b7e47cbb0cdd2ba9e576",
    "__v": 0
}
```