Welcome to MERN series as we code our service page backend. Learn about Schema, Model, Route, and Controller, and get hands-on with Atlas.

Here is our Service API

```json
[
  {
    "service": "Web Development",
    "description": "Crafting tailor-made websites and web applications.",
    "price": "$1,500 - $7,000",
    "provider": "Thapa Technical Youtube Inc."
  },
  {
    "service": "E-commerce Website Development",
    "description": "Building powerful e-commerce websites for your business.",
    "price": "$2,000 - $8,000",
    "provider": "Thapa Technical Youtube."
  },
  {
    "service": "Responsive Web Design",
    "description": "Creating visually stunning and responsive websites.",
    "price": "$1,200 - $6,000",
    "provider": "Thapa Technical Youtube Inc."
  },
  {
    "service": "Mobile App Development",
    "description": "Developing innovative and user-friendly mobile applications.",
    "price": "$2,500 - $10,000",
    "provider": "Thapa Technical Youtube."
  },
  {
    "service": "WordPress Website Development",
    "description": "Building dynamic websites using the WordPress platform.",
    "price": "$1,300 - $5,500",
    "provider": "Thapa Technical Youtube"
  },
  {
    "service": "UI/UX Design Services",
    "description": "Crafting intuitive and user-centric UI/UX designs for your projects.",
    "price": "$1,800 - $7,500",
    "provider": "Thapa Technical Youtube."
  }
```

```
]
```

1. Let's build the services model.

```js
const { Schema, model } = require("mongoose");

const serviceSchema = new Schema({
  service: { type: String, required: true },
  description: { type: String, required: true },
  price: { type: String, required: true },
  provider: { type: String, required: true },
});

const Service = new model("Service", serviceSchema);
module.exports = Service;
```

2. Create a route and add the controllers file for handling services.

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

router.route("/service").get(services);
module.exports = router;

// after
const express = require("express");
const router = express.Router();
const services = require("../controllers/service-controller");
router.route("/service").get(services);
module.exports = router;
```

3. Define the logic to retrieve services data on our services page when users visit the service route.

```js
const Service = require("../models/service-model");

const services = async (req, res) => {
  try {
    const response = await Service.find();
```

```
    if (!response) {
      res.status(404).json({ msg: "No service found" });
      return;
    }
    return res.status(200).json({ msg: "Service found", data: response });
  } catch (error) {
    console.log(`error from the server ${error}`);
  }
};

module.exports = services;
```