[1 2 4 8 7 3 5 7 8] k = 4

VS Code → Folder
(backend_dev)

Saurav Kumar
Pg-1

(1) Introduction to build REST API with Node, Express, Mongoose

npm init -y
npm i express → 4.18.2
npm i nodemon

echo app.js
→ Connect with backend

app.js

The req object represents the HTTP request and has properties for the request query string, parameters, body and HTTP header.

---

(2) Create server express.js

npm start
package.json → scripts :=

---

(3) Setup Routes and Controllers using Express Routers

Controllers → product.js
routes → products.js

→ npm run dev

Subject _____

Saurav Kumar

Page 02

get, post, put, use, listen

get →

post →

put →

product.js (routes)

---

④ Testing API using Postman & Thunderbolt

a) Download Postman - login

b) New → Collection → Product_name API

---

⑤ Connect Backed with Database

↳ Create Cluster

db (database) → Connect.js

npm i mongoose → package.json

Connect.js

~~Const mongo = require("mongoose");~~

uri ⇒ password

```
Const ConnectDB = () => {
    return mongoose.connect(uri, {
        useNewUrlParser: true,
        useUnifiedTopology: true,
    });
};

module.exports = ConnectDB;
```

Subject _____

Saurav Kumar

MON TUE WED THR FRI SAT SUN

Page 03

(6) Secure Your personal Data with DOTENV

app.js → const PORT = process.env.PORT || 5000;

↓ Environment Varible

npm i dotenv.

echo>.env

(7) Create Schema & Model (Collection & Tables) using Mongoose & ExpressJS

Product →>

→ CRUD Operation

→ product (name, description, price)

→ varient ~~(Color, Size, SKU)~~

(name)

(SKU → Stock Keeping Unit → Id)

(additional cost as compared to base product cost)

Stock Count → quantity available for purchase.

models → product.js

```
Const mongoose = require("mongoose")
Const productSchema = new mongoose.Schema({
```

```
"name": "iphone"
"description": "Mobile"
"price": 108000

"varient_name": "pro-Max"
"varient_sku": "6376 1d609 ——"
"additionalCost": "+29.000"
"stock_count": "24"
```

```
});
```

module.exports = mongoose.model("Product", productSchema)

Subject _____

Saurav kumar

Page - 84

(8)

Store Data / JSON File In DB

echo > products.json

echo > productDB.js

JSON → JavaScript Object Notation.

↓

Key-Value

Product.json

```
[
  {"name" : "iphone",
   "price" : 154,
   "features" : true,
   "Company" : "apple"
  },
  {"name" : "iphone 10"
  }
]
```

productDB.js

```
Const ConnectDB = require("./db/conn
```

↳ node productDB.js

Read Data from DB. using Express & Mongoose.

(9)

Controllers → products.js

⇒ Const myData = await Product.find({});

- Controllers → display
- db → Connectivity
- models → schema
- routes → fetch

app.js

productDB.js

Add Filteration and Searching functionality

(10)

Req. Query Props

req.body ← POST/PUT

req.query ← mostly used for Searching, Sorting, filtering, Pagination etc.

Subject

MON TUE WED THR FRI SAT SUN

(11.) Add Company filter in API and Make API work better

(12.) Add Name Advance filter in API

queryObject.name = { $regex : name , $options : "i" };

iphone → iphone, iphone10 (case insensetive)

(13) Add Sort functionality in REST API

(14) Return Specific document fields using Select

query.select

(15.) Add Pagination in REST API

→ Page
→ Page Number
→ Limits

let page = Number(req.query.page) || 1 ; (by default)

page
limit
skip = (page-1) * limit ;

(16) Host REST API Live

Railway.app

(1) push on Github

github Cli

gh repo new

git remote -v