Index.js

```
const express = require("express");
const mongoose = require("mongoose");
const dotenv = require("dotenv");
const cors = require("cors");

dotenv.config();
const app = express();

// Middleware
app.use(express.json());
app.use(cors());

// Connect to MongoDB
mongoose.connect(process.env.MONGO_URI, { useNewUrlParser: true,
useUnifiedTopology: true })
    .then(() => console.log("MongoDB Connected"))
    .catch(err => console.log(err));

app.get("/", (req, res) => res.send("API is running..."));

const PORT = process.env.PORT || 3000;
app.listen(PORT, () => console.log(`Server running on http://localhost:${PORT}`));
```

Product.js

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

const ProductSchema = new mongoose.Schema({
    title: String,
    price: Number,
    description: String,
    category: String,
    image: String,
    rating: Number,
    discount: Number,
    offerPrice: Number,
    reviews: Number
});

module.exports = mongoose.model("Product", ProductSchema);
```

User.js

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

const UserSchema = new mongoose.Schema({
   name: String,
   fatherName: String,
   dob: String,
   branch: String,
   rollNo: String,
   section: String,
   address: String,
   mobileNo: String,
   password: String
});

module.exports = mongoose.model("User", UserSchema);
```

ProductRoutes.js

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

// Add a new product
router.post("/add", async (req, res) => {
    try {
        const newProduct = new Product(req.body);
        await newProduct.save();
        res.status(201).json(newProduct);
    } catch (err) {
        res.status(500).json({ error: err.message });
    }
});

// Get all products
router.get("/all", async (req, res) => {
    try {
        const products = await Product.find();
        res.json(products);
    } catch (err) {
        res.status(500).json({ error: err.message });
    }
});

module.exports = router;
```

AuthenticationRoutes.js

```
const express = require("express");
const User = require("../models/User");
const bcrypt = require("bcrypt");
const jwt = require("jsonwebtoken");
const router = express.Router();

// Signup
router.post("/signup", async (req, res) => {
    try {
        const hashedPassword = await bcrypt.hash(req.body.password, 10);
        const newUser = new User({ ...req.body, password: hashedPassword });
        await newUser.save();
        res.status(201).json({ message: "User registered successfully!" });
    } catch (err) {
        res.status(500).json({ error: err.message });
    }
});

// Signin
router.post("/signin", async (req, res) => {
    try {
        const user = await User.findOne({ rollNo: req.body.rollNo });
        if (!user) return res.status(404).json({ error: "User not found!" });

        const isMatch = await bcrypt.compare(req.body.password, user.password);
        if (!isMatch) return res.status(401).json({ error: "Invalid credentials!" });

        const token = jwt.sign({ id: user._id }, process.env.JWT_SECRET, { expiresIn: "1h" });
        res.json({ message: "Login successful!", token });
    } catch (err) {
        res.status(500).json({ error: err.message });
    }
});

module.exports = router;
```

UpdatedIndex.js

```
const productRoutes = require("./routes/productRoutes");
const authRoutes = require("./routes/authRoutes");

app.use("/api/products", productRoutes);
app.use("/api/auth", authRoutes);
```