Practical No : 11

Aim : write a program to demonstrate organizing Express application using Router module

1 .Create a app.js

```
//app.js - main entry (clean and full version)
const express = require("express");
const morgan = require("morgan");
const bodyParser = require("body-parser");
// routes - require the actual files in `routes/`
const userRoutes = require("./routes/user");
const productRoutes = require("./routes/product");
const app = express();
app.use(morgan("dev"));
app.use(bodyParser.json());
//using routes
app.use('/api/users', userRoutes);
app.use('/api/products', productRoutes);
// error-handling middleware (must have 4 args)
app.use((err, req, res, next) => {
  console.error(err && err.stack || err);
  res.status(err?.status || 500).json({ error: err?.message || 'Internal Server Error' });
});
const PORT = process.env.PORT || 3000;
app.listen(PORT, () => {
  console.log(`Server is running on http://localhost:${PORT}`);
});
```

2. Package.json

```
 {
 "name": "express-router-full",
  "version": "1.0.0",
  "description": "",
  "license": "ISC",
  "author": "Habiba",
  "type": "commonjs",
  "main": "app.js",
  "scripts": {
   "test": "echo \"Error: no test specified\" && exit 1"
  },
  "dependencies": {
   "express.js": "^1.0.0",
   "morgan": "^1.10.1"
  }}
```

3. User.js inside the Routes

```
//routes /user.js
const express = require("express");
const router = express.Router();
const ctrl = require("../controllers/userController");

router.get("/", ctrl.listUsers);
router.get("/:id", ctrl.getUserById);
router.post("/", ctrl.createUser);
router.put("/:id", ctrl.updateUser);
router.delete("/:id", ctrl.deleteUser);

module.exports = router;
```

4. product.js inside the Routes

```
//routes/products.js
const express = require("express");
const router = express.Router();
const ctrl = require("../controllers/productController");

router.get('/', ctrl.listProducts);
router.get('/:id', ctrl.getProduct);

module.exports = router;
```

5. ProductController.js inside the Controller

```
const db = require('../Data/db');

exports.listProducts = (req, res) => {
  res.json(db.getAll('products'));
};
exports.getProduct = (req, res) => {
  const p = db.getById('products', req.params.id);
  if (!p) return res.status(404).json({ error: 'Product not found' });
  res.json(p);
};
```

6. UserController.js inside the Controller

```
const db = require('../Data/db');

exports.listUsers = (req, res) => {
  res.json(db.getAll('users'));
};
```

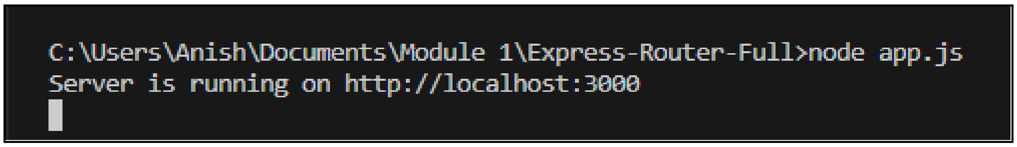
```
exports.createUser = (req, res) => {
 const newUser = db.insert('users', req.body);
 res.status(201).json(newUser);
};

exports.getUserById = (req, res) => {
 const user = db.getById('users', req.params.id);
 if (!user) return res.status(404).json({ error: 'User not found' });
 res.json(user);
};

exports.updateUser = (req, res) => {
 const updated = db.update('users', req.params.id, req.body);
 if (!updated) return res.status(404).json({ error: 'User not found' });
 res.json(updated);
};

exports.deleteUser = (req, res) => {
 const ok = db.remove('users', req.params.id);
 if (!ok) return res.status(404).json({ error: 'User not found' });
 res.status(204).send();
};
```

7. db.js inside the Data

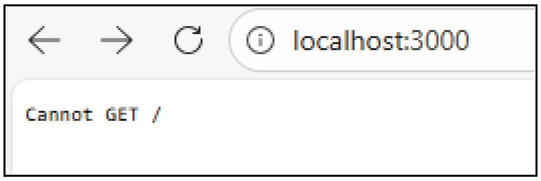
```
const store = {
 users: [
  { id: 'u1', name: 'Habiba', email: 'habiba16@gmail.com' },
  { id: 'u2', name: 'Mahek', email: 'mahek@gmail.com' },
  { id: 'u3', name: 'Zaberiya', email: 'zabii@gmail.com' }
 ],
 products: [
  { id: 'p1', name: 'Laptop', price: 1000 },
  { id: 'p2', name: 'Phone', price: 5000 },
  { id: 'p3', name: 'Tablet', price: 300 }
 ]
};

function getAll(collection) {
 return store[collection] || [];
}

function getById(collection, id) {
 return (store[collection] || []).find(o => o.id === id);
}
```

```
function generateId(prefix = '') {
  return prefix + Date.now().toString(36) + Math.floor(Math.random() * 1000).toString(36);
}

function insert(collection, obj) {
  const id = generateId(collection.slice(0, 1));
  const item = { id, ...obj };
  if (!store[collection]) store[collection] = [];
  store[collection].push(item);
  return item;
}

function update(collection, id, att) {
  const idx = (store[collection] || []).findIndex(o => o.id === id);
  if (idx === -1) return null;
  store[collection][idx] = { ...store[collection][idx], ...att };
  return store[collection][idx];
}

function remove(collection, id) {
  const idx = (store[collection] || []).findIndex(o => o.id === id);
  if (idx === -1) return false;
  store[collection].splice(idx, 1);
  return true;
}

module.exports = { getAll, getById, insert, update, remove };
```

Output:-



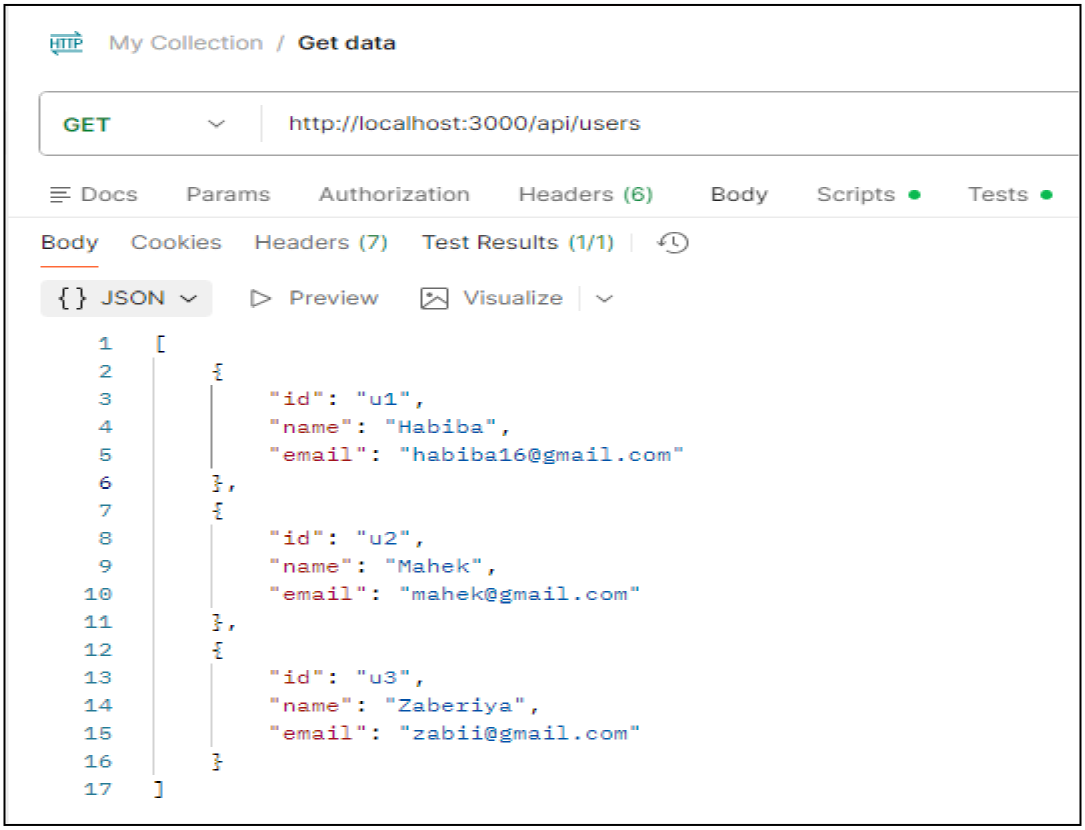
```
C:\Users\Anish\Documents\Module 1\Express-Router-Full>node app.js
Server is running on http://localhost:3000
```



```
Cannot GET /
```

Post man :-

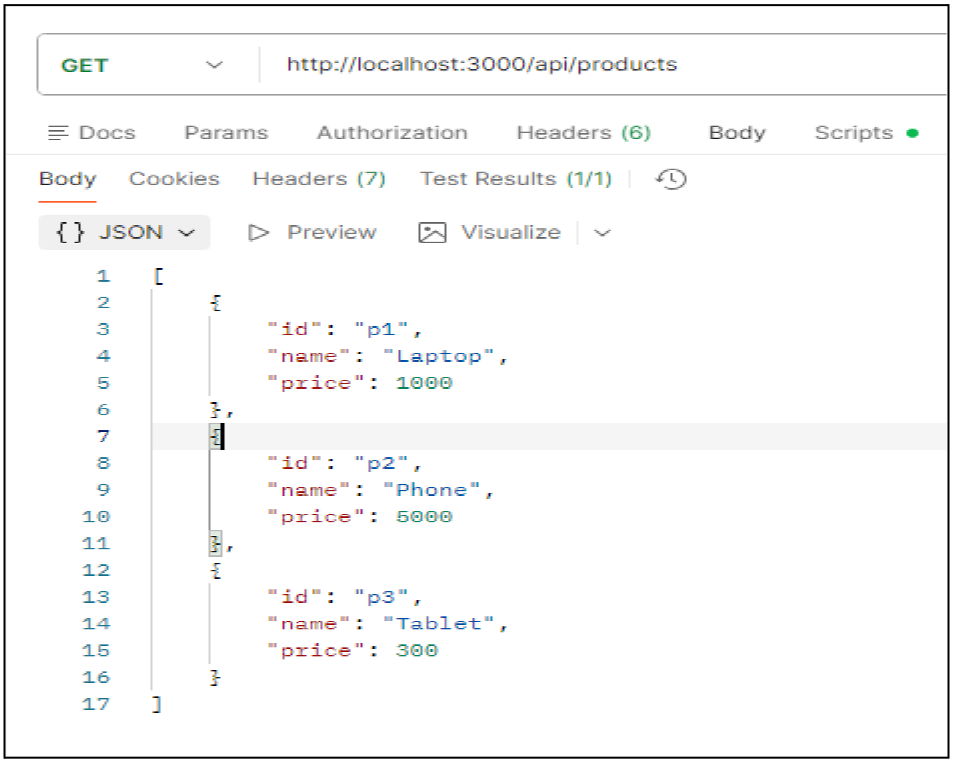
1. Users data



GET http://localhost:3000/api/users

```
[
    {
        "id": "u1",
        "name": "Habiba",
        "email": "habiba16@gmail.com"
    },
    {
        "id": "u2",
        "name": "Mahek",
        "email": "mahek@gmail.com"
    },
    {
        "id": "u3",
        "name": "Zaberiya",
        "email": "zabii@gmail.com"
    }
]
```

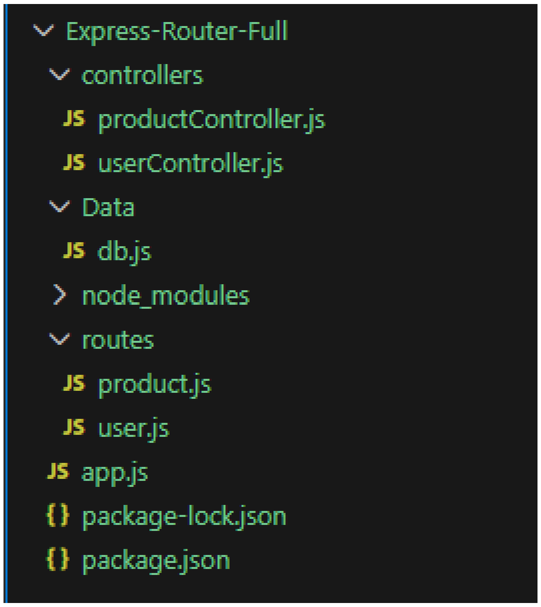
2. Products data



GET http://localhost:3000/api/products

```
[
    {
        "id": "p1",
        "name": "Laptop",
        "price": 1000
    },
    {
        "id": "p2",
        "name": "Phone",
        "price": 5000
    },
    {
        "id": "p3",
        "name": "Tablet",
        "price": 300
    }
]
```

Folder Structure:-



```
Express-Router-Full
  controllers
    productController.js
    userController.js
  Data
    db.js
  node_modules
  routes
    product.js
    user.js
  app.js
  package-lock.json
  package.json
```