Additional Activities

Activity 16 – PostIT App – Environment Variables

The objectives of this activity is to:

a) Create and utilize environment variables on the server and client side for secure and dynamic configuration.

Environment variables are key-value pairs that we use in an application to specify the configuration settings. They are saved outside of the program, and we use this to store sensitive data that should not be exposed such as when uploading to version control or during deployment. It also helps to manage the application easily but allows us to change the configuration settings in one location.

1. In the server folder, create a new file: **.env.** No need to specify a filename, just the file extension of **.env.**
2. Create environment variables such as the following:

```
DB_HOST=localhost
DB_USER=jasminetumulak
DB_PASSWORD=oman12345
DB_CLUSTER=cluster0.lvic91v.mongodb.net
DB_NAME=postITDb
PORT = 3001
CLIENT_URL=http://localhost:3000
```

3. When the application is launched, the environment variables are loaded in the **process.env** which is a Node.js global object with allows you to access environment variables.
   For example: **process.env.DB_HOST.**
   To access your **process.env,** install in the server the dotenv package.

```
npm install dotenv
```

4. In the server folder, create a new file: **config.js.** Import the **dotenv** package. Create variables to retrieve the values from the environment variables. Make sure to export all of them.

```
import dotenv from "dotenv";

dotenv.config(); //retrieve the environment variables.
export const PORT = process.env.PORT;
export const DB_USER = process.env.DB_USER;
export const DB_PASSWORD = process.env.DB_PASSWORD;
export const DB_NAME = process.env.DB_NAME;
export const DB_CLUSTER = process.env.DB_CLUSTER;
export const CLIENT_URL = process.env.CLIENT_URL;
```

5. In **index.js**, import all variables from the **config.js** file. Code below means that all named exports from **config.js** will be accessible as properties of the ENV object.

```
import * as ENV from "./config.js";
```

For example: **ENV.DB_HOST**

6. Update your connection string to make it dynamic by using the environment variables. For example:

Previous connection string:

```
const connectString =
  "mongodb+srv://jasminetumulak:oman12345@cluster0.lvic91v.mongodb.net
/postITDb?retryWrites=true&w=majority&appName=Cluster0";
```

Change to:

```
const connectString =
`mongodb+srv://${ENV.DB_USER}:${ENV.DB_PASSWORD}@${ENV.DB_CLUSTER}/${ENV.DB_NAME}?r
etryWrites=true&w=majority&appName=Cluster0`;
```

7. In preparation for deployment, update:

```
//Middleware
const corsOptions = {
  origin: ENV.CLIENT_URL, //client URL local
  methods: "GET,HEAD,PUT,PATCH,POST,DELETE",
  credentials: true, // Enable credentials (cookies, authorization
headers, etc.)
};
app.use(cors(corsOptions));
```

8. Change the port to the environment variable.

```
const port = ENV.PORT || 3001;
app.listen(port, () => {
  console.log(`You are connected at port: ${port}`);
});
```

9. In **client/ folder,** create a new file: **.env**. Make sure it is in the same location as **package.json**. All environment variable names must start with **REACT_APP_** to be accessible in React.
   For example:

```
#Server URL-development (local server)
REACT_APP_SERVER_URL=http://localhost:3001
```

10. In **client/src** folder, create a new file: config.js.

```
export const SERVER_URL = process.env.REACT_APP_SERVER_URL;
```

11. In **client/src/Features/UserSlice.js and in the PostSlice.js**.

```
import * as ENV from "../config";
```

12. Change the codes which explicitly specify the server URL and change it to the environment variable SERVER_URL. Make sure to change the "" to ``.

Change this code:

```
const response = await axios.post("http://localhost:3001/registerUser"
```

To this:

```
const response = await axios.post(`${ENV.SERVER_URL}/registerUser`
```

Do this for all Axios requests.