Assignment - 13

Frontend & Backend Integration.

→ In web development, frontend & backend are distint layer of an application that communicate to provide seamless user experience. The frontend is responsible for the user interface and uses interface, while the backend handle business, logic, database, interactions and server side processing.

Restful API :-

A restful API is an API adheres to the principles of REST (Representational State Transfer). REST is an architectural style for designed networked applications, relying on a stateless, client-server communication protocal typically HTTP.

Principles of REST :-

① Client - Server Architecture :- Separates the user interface concerns from data storage concerns, improving the scalability & probability of an application.

② Statelessness :- Each request from a client do a server must contain all the information needed to understand & process the request.

③ Cachability :- Responses must defines themselves as cachebility or non-cacheable to improve client-side performance.

④ Uniform Interface :- Simplifies & decouple the architecture, allowing each part to evolve independently.

⑤ Layered System :- The client cannot tell whether it is connected directly to the end server or an intermediary, improving scalability & security.

⑥ Code on Demand :- Servers can extend client functionality by transferring executable code.

HTTP Methods :-

① GET : Retrieve data from Server.
② POST : Send data to server to create
③ PUT : Update an existing resource
④ DELETE : Remove a resource.
⑤ PATCH :- Apply partial modifications.

Making API calls from FrontEnd :-

① Fetch :-
```
fetch ('http://api.example.com/data')
.then (response => response.json())
.then (data => console.log (data))
.catch (error => console.log.error ('error', error))
```

② Axios :-
```
axios.get ('http://api.example.com/data')
.then (response => {
    console.log (response.data);
})
.catch (error => {
    console.log.error ('Error', error);
});
```

Create the Server :-

```
① app.set ('/api/data', (req, res) => {
        res.json(data);
    });

② app.listen(port, () => {
        console.log('server running at ${port}');
    });
```