PART 1: `deletePost` — Line-by-Line Mapping

✅ Frontend Code (jo tu pehle diya tha):

js
CopyEdit

```js
const deletePostHandler = async () => {
  const res = await
axios.delete(`http://localhost:8080/api/v1/post/delete/${post._id}
`, {
    withCredentials: true
  });
}
```

🖇 Ye frontend backend ke kis line ko call karta hai?

🔍 Backend Function: `deletePost`

js
CopyEdit

```js
export const deletePost = async (req, res) => {
```

- ✅ Ye controller call hota hai jab frontend `axios.delete(...)` karta hai.

🔢	Backend Code	🖇 Frontend Se	📖 Explanation
1	`const postId = req.params.id;`	`post._id` URL se jaata hai	URL me `/delete/${post._id}` tha
2	`const authorId = req.id;`	`withCredentials: true`	Cookie/session se login user ka ID milta hai
3	`const post = await Post.findById(postId);`	❌ (internal)	DB se post fetch hoti hai

4	`if(!post)`	❌	Agar post hi na mile to error
5	`if(post.author.toString() !== authorId)`	❌	Sirf post ka owner hi delete kar sakta hai
6	`await Post.findByIdAndDelete(postId);`	🔥 Actual delete call	
7	`let user = await User.findById(authorId);`	❌	User ka data fetch karte ho
8	`user.posts = user.posts.filter(...)`	❌	User ki post list se is post ka ID hata diya
9	`await user.save();`	❌	Updated user save hota hai
10	`await Comment.deleteMany({ post: postId });`	❌	Us post ke sabhi comments delete ho jaate hain
11	`res.status(200).json({... })`	`res.data.message` (frontend)	Ye message toast me dikhta hai frontend pe

🔢	Backend Code	🔗 Frontend Se	📖 Explanation
1	`const postId = req.params.id;`	`post._id` from URL	Konsi post bookmark karni hai
2	`const authorId = req.id;`	`withCredentials: true`	User ID cookie/session se aayi
3	`const post = await Post.findById(postId);`	❌	Post exist karti hai ya nahi
4	`if (!post)`	❌	Post na ho to error bhejta hai
5	`const user = await User.findById(authorId);`	❌	User ka data fetch

6	`if (user.bookmarks.includes(post._id))`	❌	Already bookmarked?
7	`$pull:{bookmarks:post._id}`	✅ Agar already bookmarked hai to hata deta hai	
8	`$addToSet:{bookmarks:post._id}`	✅ Nahi hai to add karta hai	
9	`await user.save();`	❌	Updated user DB me save hota hai
10	`res.status(200).json({ message: ... })`	`toast.success(...)` frontend me show hota hai	

🔢	Frontend Action	🔗 Backend Logic Triggered	📖 Explanation
1	`axios.delete(...)`	`deletePost(req,res)`	Delete post
2	`post._id` in URL	`req.params.id`	Backend ko post ID milti hai
3	`withCredentials: true`	`req.id`	Authenticated user ID backend me aati hai
4	`axios.get(...bookmark)`	`bookmarkPost(req,res)`	Bookmark toggle
5	`toast.success(...)`	`res.status().json({...})`	Server response UI me dikhaya jaata hai

🧩 Step	🖥 Frontend Code	🖥 Backend Code	🔗 Connection / Explanation

1	`axios.delete("http://localhost:8080/api/v1/post/delete/${post._id}")`	`export const deletePost = async (req,res)`	Frontend `DELETE` request se backend controller call hota hai
2	`post._id`	`const postId = req.params.id;`	Post ka ID URL me jaata hai aur backend me `req.params.id` ban jaata hai
3	`withCredentials: true`	`const authorId = req.id;`	Login user ki ID cookies/session se backend me `req.id` me milti hai
4	❌	`const post = await Post.findById(postId);`	Backend me DB se post dhoondi ja rahi hai
5	❌	`if(post.author.toString() !== authorId)`	Yahan check ho raha hai ki sirf post ka owner hi delete kar sake

6	❌	`await Post.findByIdAndDelete(postId);`	Actual DB se post delete hoti hai
7	❌	`user.posts = user.posts.filter(...)`	User ke profile se bhi post remove hoti hai
8	❌	`await Comment.deleteMany({ post: postId });`	Post se related comments bhi delete ho jaate hain
9	`toast.success(res.data.message)`	`res.status(200).json({ message: 'Post deleted' })`	Backend ka response frontend me toast ke through show hota hai
10	`dispatch(setPosts(...))`	❌	Frontend state se bhi post hata di jaati hai

🧩 **Step**	🖥️ **Frontend Code**	🖥️ **Backend Code**	🔗 **Connection / Explanation**
1	`axios.get("http://localhost:8080/api/v1/post/${post._id}/bookmark")`	`export const bookmarkPost = async (req,res)`	Frontend ka GET request backend

			controller ko call karta hai
2	`post._id`	`const postId = req.params.id;`	Post ID URL me jaata hai, backend me `params` se milta hai
3	`withCredentials: true`	`const authorId = req.id;`	User ID session/cookie se backend me aata hai
4	❌	`const post = await Post.findById(postId);`	Backend post existence check karta hai
5	❌	`const user = await User.findById(authorId);`	User data DB se fetch hota hai
6	❌	`if (user.bookmarks.includes(post._id))`	Bookmark hai ya nahi check ho raha
7	❌	`$pull:{bookmarks: post._id}` ya `$addToSet:{bookmarks: post._id}`	Agar already bookmarked hai to remove, nahi to add

8	❌	`await user.save();`	Updated user DB me save hota hai
9	`toast.success(res.data.message)`	`res.status(200).json({ message: 'Post bookmarked' })`	Response message frontend me toast ke through show hota hai