

TEP 1: FRONTEND CODE — likeOrDislikeHandler

js

CopyEdit

```
const likeOrDislikeHandler = async () => {
```

- ◆ Function start: Jab user like/dislike button pe click karega, yeh function chalega.

js

CopyEdit

```
const action = liked ? 'dislike' : 'like';
```

- ◆ Agar post already liked hai (`liked === true`), toh ab **dislike** hoga, warna **like**.

 `action = "like" ya "dislike"`

js

CopyEdit

```
const res = await
axios.get(`http://localhost:8080/api/v1/post/${post._id}/${action}`
, {
  withCredentials: true
});
```

- ◆ Backend ko GET request bheji ja rahi hai.
- ◆ URL example:

bash

CopyEdit

```
http://localhost:8080/api/v1/post/12345/like
```

- ◆ `withCredentials: true` → user ki authentication cookie bhi jaayegi.

 **Yeh kis controller ko hit karega?**

- Agar URL me `/like` hai → `likePost`
- Agar `/dislike` hai → `dislikePost`

js

CopyEdit

```
if (res.data.success) {
```

- ◆ Agar backend ne `success: true` bheja, toh hum frontend UI update karenge.

js

CopyEdit

```
const updatedLikes = liked ? postLike - 1 : postLike + 1;
setPostLike(updatedLikes);
setLiked(!liked);
```

- ◆ UI update:

- Agar already liked tha, count -1
- Agar nahi tha, count +1
- `setLiked(!liked)` → toggle state

js

CopyEdit

```
const updatedPostData = posts.map(p =>
  p._id === post._id
    ? {
        ...p,
        likes: liked
          ? p.likes.filter(id => id !== user._id)
          : [...p.likes, user._id]
      }
    : p
);
```

- ◆ Post ki `likes` array ko update kiya:

- Agar dislike ho raha: current user ID remove
- Agar like ho raha: current user ID add

js

CopyEdit

```
dispatch(setPosts(updatedPostData));
```

- ◆ Redux state me updated posts bhej diye, taaki UI har jagah same ho.

js

CopyEdit

```
toast.success(res.data.message);
```

- ◆ Backend se aayi message string ko success toast me dikhaya.

js

CopyEdit

```
} catch (error) {  
  console.log(error);  
}  
}
```

- ◆ Agar koi error aata hai (server error, auth issue), console me dikha diya.



SUMMARY: FRONTEND

| Kaam | Explanation |
|--------------------|------------------------------------------------|
| Action decide | liked hai toh "dislike", warna "like" |
| Axios call | <code>/post/:id/:action</code> backend request |
| Backend success pe | Like count & icon toggle |
| Redux update | Post ke likes array me update |
| Toast message | "Post liked" ya "Post disliked" |

✂ STEP 2: BACKEND CODE — likePost & dislikePost

◆ LIKE CONTROLLER: likePost

js

CopyEdit

```
export const likePost = async (req, res) => {
  try {
    const likeKrneWalaUserKiId = req.id;
    const postId = req.params.id;
```

◆ Auth middleware se `req.id` mila (current user ki ID)

◆ URL se `postId` mila

js

CopyEdit

```
    const post = await Post.findById(postId);
    if (!post) return res.status(404).json({ message: 'Post not found', success: false });
```

◆ DB se post dhoondha

◆ Agar post nahi mila toh error response bhej diya

js

CopyEdit

```
    await post.updateOne({ $addToSet: { likes: likeKrneWalaUserKiId } });
    await post.save();
```

◆ `$addToSet`:

- Like krne wale user ki ID ko `likes` array me daal diya
 - Duplicate avoid hota hai
-

js

CopyEdit

```
const user = await
User.findById(likeKrneWalaUserKiId).select('username
profilePicture');
```

◆ User ki info nikali (notification ke liye)

js

CopyEdit

```
const postOwnerId = post.author.toString();
if(postOwnerId !== likeKrneWalaUserKiId){
```

◆ Agar user apna hi post like kar raha hai, notification nahi bhejna

js

CopyEdit

```
const notification = {
  type:'like',
  userId:likeKrneWalaUserKiId,
  userDetails:user,
  postId,
  message:'Your post was liked'
}
```

◆ Notification object ready

js

CopyEdit

```
const postOwnerSocketId = getReceiverSocketId(postOwnerId);
io.to(postOwnerSocketId).emit('notification', notification);
```

◆ Socket.io se real-time notification send kiya

js

CopyEdit

```
return res.status(200).json({message:'Post liked',
success:true});
} catch (error) {}
```

```
}
```

◆ Response wapas frontend ko bhej diya: success & message

◆ DISLIKE CONTROLLER: **dislikePost**

js

CopyEdit

```
await post.updateOne({ $pull: { likes: likeKrneWalaUserKiId } });
```

◆ **\$pull** → **likes** array me se user ID hata di gayi (dislike)

Baaki logic same hai → Notification emit, response send



Frontend Code Line

```
1  const action = liked ? 'dislike' :  
   'like';
```

```
2  axios.get('http://localhost:8080/api  
   /v1/post/${post._id}/${action}', {  
   withCredentials: true });
```



Connects to Backend Code Line

✗ No backend call here

```
→ const postId =  
   req.params.id;const  
   likeKrneWalaUserKi  
   Id = req.id;
```








Kya hota hai

Bas
yeh
decide
karta
hai ki
like
bhejna
hai ya
dislike

Fronte
nd
GET
request
t
bhejta
hai,
backe
nd
URL
aur

| | | | |
|---|-------------|----------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| | | | login user ID fetch karta hai |
| 3 | (same line) | <pre> → const post = await Post.findById(post Id); </pre> | Post databa se se find hoti hai |
| 4 | (same line) | <pre> → If action == like → likePost() runs If action == dislike → dislikePost() runs </pre> | Contro ller decide hota hai kis action pe |
| 5 | --- | <pre> await post.updateOne({ \$addToSet: { likes: likeKrneWalaUserKi Id } }); </pre> | User ki ID post ke likes me add hoti hai |
| 6 | --- | <pre> await post.updateOne({ \$pull: { likes: likeKrneWalaUserKi Id } }); </pre> | (Agar dislike) User ki ID remov e hoti hai likes se |
| 7 | --- | <pre> return res.status(200).js on({message:'Post liked', success:true}); </pre> | Backe nd succe ss respon se |

| | | | |
|--------|---------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------|
| | | | bhejta hai |
| 8 | <code>if(res.data.success){ ... }</code> |  Connects to previous line | Frontend check karta hai success response aaya ya nahi |
| 9 | <code>setPostLike(updatedLikes);</code> |  No backend link | Frontend pe like count UI update hoti hai |
| 1 0 | <code>setLiked(!liked);</code> |  No backend link | Like/dislike icon UI me toggle hota hai |
| 1 1 | <code>dispatch(setPosts(updatedPostData));</code> |  No backend link | Redux me updated post store hota hai |
| 1 2 | <code>toast.success(res.data.message);</code> |  Backend se aayi message frontend toast me dikhai jaati hai | Notification text show hota hai |