

```

1 import React, { useState, useCallback } from 'react';
2 import { Card, Button, Avatar, Popover, List, Comment } from 'antd';
3 import PropTypes from 'prop-types';
4 import { RetweetOutlined, HeartTwoTone, HeartOutlined, MessageOutlined, EllipsisOutlined }
  from '@ant-design/icons';
5 import styled from 'styled-components';
6 import Link from 'next/link';
7 import { useSelector } from 'react-redux';
8
9 import CommentForm from './CommentForm';
10 import PostCardContent from './PostCardContent';
11 import PostImages from './PostImages';
12 import FollowButton from './FollowButton';
13
14 const CardWrapper = styled.div`
15   margin-bottom: 20px;
16 `;
17
18 const PostCard = ({ post }) => {
19   const [commentFormOpened, setCommentFormOpened] = useState(false);
20   const id = useSelector((state) => state.user.me && state.user.me.id);
21
22   const [liked, setLiked] = useState(false);
23
24   const onToggleLike = useCallback(() => {
25     setLiked((prev) => !prev);
26   }, []);
27
28   const onToggleComment = useCallback(() => {
29     setCommentFormOpened((prev) => !prev);
30   }, []);
31
32   return (
33     <CardWrapper key={post.id}>
34       <Card
35         cover={post.Images[0]} && <PostImages images={post.Images} />
36         actions={[
37           <RetweetOutlined key="retweet" />,
38           liked
39             ? <HeartTwoTone twoToneColor="#eb2f96" key="heart" onClick={onToggleLike} />
40             : <HeartOutlined key="heart" onClick={onToggleLike} />,
41           <MessageOutlined key="message" onClick={onToggleComment} />,
42           <Popover
43             key="ellipsis"
44             content={
45               <Button.Group>
46                 {id && post.User.id === id
47                   ? (
48                     <>
49                       <Button>수정</Button>
50                       <Button type="danger">삭제</Button>
51                     </>
52                   )
53                   : <Button>신고</Button>}
47                 </Button.Group>
48               )}
49             >
50               <EllipsisOutlined />
51             </Popover>,

```

```
59     ]}
60     extra={<FollowButton post={post} />}
61   >
62     <Card.Meta
63       avatar={<Avatar>{post.User.nickname[0]}</Avatar>}
64       title={post.User.nickname}
65       description={<PostCardContent postData={post.content} />}
66     />
67   </Card>
68   {commentFormOpened && (
69     <
70       <CommentForm post={post} />
71       <List
72         header={` ${post.Comments.length} 댓글`}
73         itemLayout="horizontal"
74         dataSource={post.Comments}
75         renderItem={(item) => (
76           <li>
77             <Comment
78               author={item.User.nickname}
79               avatar={
80                 <Link href={{ pathname: '/user', query: { id: item.User.id } }} as=
81                 ` /user/${item.User.id}`>
82                   <a><Avatar>{item.User.nickname[0]}</Avatar></a>
83                 </Link>
84               )}
85               content={item.content}
86             />
87           </li>
88         )}
89       </>
90     )}
91   </CardWrapper>
92 );
93 };
94
95 PostCard.propTypes = {
96   post: PropTypes.shape({
97     id: PropTypes.number,
98     User: PropTypes.object,
99     content: PropTypes.string,
100    createdAt: PropTypes.object,
101    Comments: PropTypes.arrayOf(PropTypes.any),
102    Images: PropTypes.arrayOf(PropTypes.any),
103  }),
104 };
105
106 export default PostCard;
107
```