Step by step explanation for Reviews component

1. The Reviews component takes in a gigId prop. This is used to fetch the reviews for a specific gig.
2. The useQueryClient hook is used to create a queryClient object. This object is used to interact with the React Query library.
3. The useQuery hook is used to fetch data from the server. In this case, it fetches the reviews for the specified gigId. The data is stored in the data variable and can be accessed in the component.
4. The useMutation hook is used to create a mutation object. This object is used to send new reviews to the server. When a new review is successfully posted to the server, the "reviews" query is invalidated using the queryClient.invalidateQueries(["reviews"]) method. This means that the data for the "reviews" query will be refetched from the server.
5. The handleSubmit function is called when the form in the component is submitted. It gets the values of the form inputs and uses them to create a new review object. Then it calls the mutate function on the mutation object and passes in the new review object.
6. The component returns some JSX that displays a list of reviews and a form to add a new review. The list of reviews is generated by mapping over the data variable and rendering a Review component for each review.

Step by step explanation for Review component

1. The Review component takes in a review prop. This is an object that contains information about a review, such as the user who wrote it and the star rating.
2. The useQuery hook is used to fetch data from the server. In this case, it fetches information about the user who wrote the review using the review.userId value. The data is stored in the data variable and can be accessed in the component.
3. The component returns some JSX that displays information about the review and the user who wrote it. It shows the user’s profile picture, username and country. It also shows the star rating of the review and its description.
4. If the data is still being fetched from the server or if there was an error fetching the data, a loading or error message is displayed instead.

How new review is added?

When the user clicks the submit button, the handleSubmit function is called. This function gets the values of the form inputs and uses them to create a new review object. Then it calls the mutate function on the mutation object and passes in the new review object.

When the mutate function is called, it triggers the mutationFn specified in the useMutation hook. In this case, the mutationFn sends a POST request to the /reviews endpoint with the new review data using the newRequest object.

After the new review is successfully posted to the server, the onSuccess function specified in the useMutation hook is called. In this case, it invalidates the "reviews" query using the queryClient.invalidateQueries(["reviews"]) method. This means that the data for the "reviews" query will be refetched from the server and the list of reviews displayed in the component will be updated to include the new review.

Gigs component step by step process

1. The Gigs component uses several hooks to manage its state and behavior. The useState hook is used to create open, setOpen, sort, and setSort variables. These are used to control whether a menu is open and how the data is sorted. The useRef hook is used to create minRef and maxRef variables. These are used to access the values of the minimum and maximum price inputs in the component.
2. The useLocation hook is used to get the current location object from the React Router. The search property of this object contains the query string of the current URL.
3. The useQuery hook is used to fetch data from the server. In this case, it fetches a list of gigs based on the current query string, minimum and maximum price values, and sort order. The data is stored in the data variable and can be accessed in the component.
4. The resort function is called when a sort option is clicked. It sets the value of the sort variable and closes the menu.
5. The useEffect hook is used to refetch the data from the server whenever the value of the sort variable changes.
6. The apply function is called when the apply button is clicked. It closes the menu and refetches the data from the server.
7. The component returns some JSX that displays a list of gigs and a menu to filter and sort them. The list of gigs is generated by mapping over the data variable and rendering a GigCard component for each gig.

Gigs component step by step process

1. The Gig component is defined as a functional component that takes no props.
2. The useParams hook from the react-router-dom library is used to get the id of the gig from the URL.
3. The useQuery hook from the react-query library is used to fetch data about the gig from an API. The query key is set to ["gig"] and the query function sends a GET request to /gigs/${id} using the newRequest object. The response data is returned by the query function.
4. Another useQuery hook is used to fetch data about the seller of the gig. This query is only enabled if data?.userId is truthy, meaning that the first query has returned data with a userId property. The query key is set to ["user"] and the query function sends a GET request to /users/${data?.userId} using the newRequest object. The response data is returned by the query function.
5. The component returns some JSX that displays a loading message if either query is still loading, an error message if either query has returned an error, or a container with information about the gig and its seller if both queries have returned data successfully.