App.js

```jsx
// Context

import { v4 as uuidv4 } from "uuid"
import { BrowserRouter as Router, Routes, Route } from "react-router-dom"
import { useState } from "react";
import Header from "./components/Header";
import FeedbackList from "./components/FeedbackList";
import FeedbackData from "./data/FeedbackData";
import FeedbackStats from "./components/FeedbackStats";
import FeedbackForm from "./components/FeedbackForm";
import AboutPage from "./pages/AboutPage"
import AboutIconLink from "./AboutIconLink";
import { FeedbackProvider } from "./context/FeedbackContext"
// We didn't use export default, Thats why {FeedbackProvider}

function App() {
  const [feedback, setFeedback] = useState(FeedbackData);

  const deleteFeedback = (id) => {
    if (window.confirm("Are you sure you want to delete?")) {
      setFeedback(feedback.filter((item) => item.id !== id));
    }

  }

  const addFeedback = (newFeedback) => {
    newFeedback.id = uuidv4()
    setFeedback([newFeedback, ...feedback])
  }

  return (
    <FeedbackProvider>
      <Router>
        <Header />
        <div className="container">
          <Routes>
            <Route exact path="/" element={
              <>
                <FeedbackForm handleAdd={addFeedback} />
                <FeedbackStats feedback={feedback} />
                <FeedbackList feedback={feedback} handleDelete={deleteFeedback} />

              </>
            }>
            </Route>

            <Route path="/about" element={<AboutPage />} />
          </Routes>
          <AboutIconLink />
        </div>
      </Router>
    </FeedbackProvider>
  );
}

export default App;
```

FeedbackContext.jsx

```jsx
import { createContext, useState } from "react";

const FeedbackContext = createContext();

export const FeedbackProvider = ({ children }) => {
  const [feedback, setFeedback] = useState([
      // We set the this obj for data to all files. Children means all values inside of
FeedbackContext, we send all of them for use at another file

    {
      id: 1,
      text: "This item from context",
      rating: 10,
    },
  ]);
  return (
    <FeedbackContext.Provider
      value={{
        feedback,
      }}
    >
      {children}
    </FeedbackContext.Provider>
  );
};

export default FeedbackContext;
```

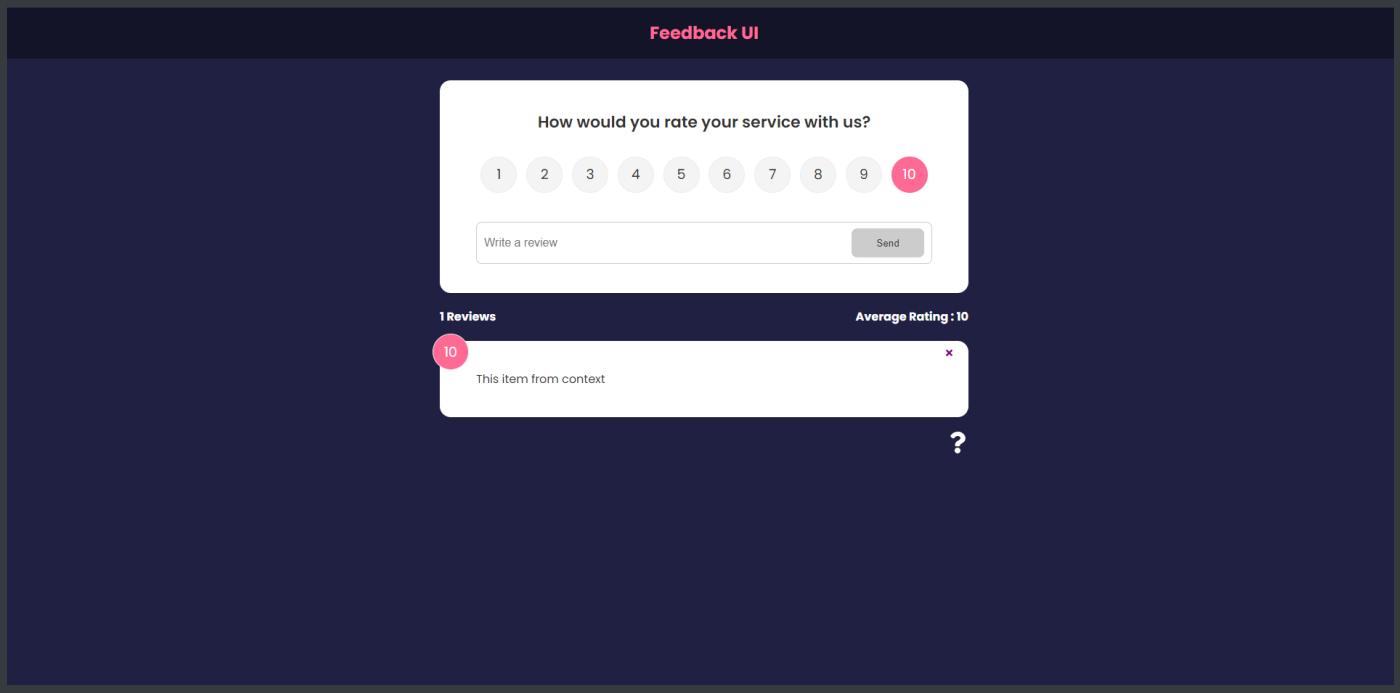
```jsx
import { v4 as uuidv4 } from "uuid"
import { BrowserRouter as Router, Routes, Route } from "react-router-dom"
import { useState } from "react";
import Header from "./components/Header";
import FeedbackList from "./components/FeedbackList";
import FeedbackData from "./data/FeedbackData";
import FeedbackStats from "./components/FeedbackStats";
import FeedbackForm from "./components/FeedbackForm";
import AboutPage from "./pages/AboutPage"
import AboutIconLink from "./AboutIconLink";
import { FeedbackProvider } from "./context/FeedbackContext"


function App() {
  const [feedback, setFeedback] = useState(FeedbackData);

  const deleteFeedback = (id) => {
    if (window.confirm("Are you sure you want to delete?")) {
      setFeedback(feedback.filter((item) => item.id !== id));
    }

  }

  const addFeedback = (newFeedback) => {
    newFeedback.id = uuidv4()
    setFeedback([newFeedback, ...feedback])
  }

  return (
    <FeedbackProvider>
      <Router>
        <Header />
        <div className="container">
          <Routes>
            <Route exact path="/" element={
              <>
                <FeedbackForm handleAdd={addFeedback} />
                <FeedbackStats /> // We delete props here ()
                <FeedbackList handleDelete={deleteFeedback} />

              </>
            }>
            </Route>

            <Route path="/about" element={<AboutPage />} />
          </Routes>
          <AboutIconLink />
        </div>
      </Router>
    </FeedbackProvider>
  );
}

export default App;
```

FeedbackList.jsx

```jsx
import { motion, AnimatePresence } from "framer-motion";
import { useContext } from "react";
import FeedbackItem from "./FeedbackItem";
import FeedbackContext from "../context/FeedbackContext";

function FeedbackList({ handleDelete }) {
  const { feedback } = useContext(FeedbackContext);
    // When we use Context, we can delete all props and props.type we use for feedback

  if (!feedback || feedback.length === 0) {
    return <p>No Feedback Yet!</p>;
  }

  return (
    <div className="feedback-list">
      <AnimatePresence>
        {feedback.map((item) => (
          <motion.div
            key={item.id}
            initial={{ opacity: 0 }}
            animate={{ opacity: 1 }}
            exit={{ opacity: 0 }}
          >
            <FeedbackItem
              key={item.id}
              item={item}
              handleDelete={handleDelete}
            />
          </motion.div>
        ))}
      </AnimatePresence>
    </div>
  );
}

export default FeedbackList;
```

FeedbackStats.jsx

```jsx
import { useContext } from "react";
import FeedbackContext from "../context/FeedbackContext";

function FeedbackStats({}) {
  const { feedback } = useContext(FeedbackContext);
    // We did Same thing like above.

  // Calculate Ratings Avg
  let average =
    feedback.reduce((acc, cur) => {
      return acc + cur.rating;
    }, 0) / feedback.length;

  average = average.toFixed(1).replace(/[.,]0$/, "");

  return (
    <div className="feedback-stats">
      <h4>{feedback.length} Reviews</h4>
      <h4>Average Rating : {isNaN(average) ? 0 : average}</h4>
    </div>
  );
}

export default FeedbackStats;
```



This 1 review comes from FeedbackContext.jsx. We'll change it later.

App.JS

```
import { BrowserRouter as Router, Routes, Route } from "react-router-dom"
import Header from "./components/Header";
import FeedbackList from "./components/FeedbackList";
import FeedbackStats from "./components/FeedbackStats";
import FeedbackForm from "./components/FeedbackForm";
import AboutPage from "./pages/AboutPage"
import AboutIconLink from "./AboutIconLink";
import { FeedbackProvider } from "./context/FeedbackContext"


function App() {

  return (
    <FeedbackProvider>
      <Router>
        <Header />
        <div className="container">
          <Routes>
            <Route exact path="/" element={
              <>
                <FeedbackForm /> // We deleted all props from our components
                <FeedbackStats />
                <FeedbackList />

              </>
            }>
            </Route>

            <Route path="/about" element={<AboutPage />} />
          </Routes>
          <AboutIconLink />
        </div>
      </Router>
    </FeedbackProvider>
  );
}

export default App;
```

```jsx
import { createContext, useState } from "react";
import { v4 as uuidv4 } from "uuid";

const FeedbackContext = createContext();

export const FeedbackProvider = ({ children }) => {
  const [feedback, setFeedback] = useState([
    {
      id: 1,
      text: "This is feedback item 1",
      rating: 10,
    },
    {
      id: 2,
      text: "This is feedback item 2",
      rating: 9,
    },
    {
      id: 3,
      text: "This is feedback item 3",
      rating: 7,
    },
  ]);
  const deleteFeedback = (id) => { //  We moved our functions to FeedbackContext
    if (window.confirm("Are you sure you want to delete?")) {
      setFeedback(feedback.filter((item) => item.id !== id));
    }
  };

  const addFeedback = (newFeedback) => { // We moved our functions to FeedbackContext
    newFeedback.id = uuidv4();
    setFeedback([newFeedback, ...feedback]);
  };

  return (
    <FeedbackContext.Provider
      value={{ // We set them as value to here.
        feedback,
        deleteFeedback,
        addFeedback,
      }}
    >
      {children}
    </FeedbackContext.Provider>
  );
};

export default FeedbackContext;
```

FeedbackForm.jsx

```jsx
import { useState, useContext } from "react";
import Card from "./shared/Card";
import Button from "./shared/Button";
import RatingSelect from "./RatingSelect";
import FeedbackContext from "../context/FeedbackContext";

function FeedbackForm() {
```

```
  const [text, setText] = useState("");
  const [rating, setRating] = useState(10);
  const [btnDisabled, setBtnDisabled] = useState(true);
  const [message, setMessage] = useState("");

  const { addFeedback } = useContext(FeedbackContext);
    // We import useContext upside and we catch addFeedback from FeedbackContext with useContext
above

  const handleTextChange = (e) => {
    if (text === "") {
      setBtnDisabled(true);
      setMessage(null);
    } else if (text !== "" && text.trim().length <= 10) {
      setMessage("Text must be at least 10 characters long");
      setBtnDisabled(true);
    } else {
      setMessage(null);
      setBtnDisabled(false);
    }
    setText(e.target.value);
  };

  const handleSubmit = (e) => {
    e.preventDefault();
    if (text.trim().length > 10) {
      const newFeedback = {
        text,
        rating,
      };

      addFeedback(newFeedback); // We change the name of func. It comes from FeedbackContext

      setText("");
    }
  };

  return (
    <Card>
      <form onSubmit={handleSubmit}>
        <h2>How would you rate your service with us?</h2>
        <RatingSelect select={(rating) => setRating(rating)} />
        <div className="input-group">
          <input
            onChange={handleTextChange}
            type="text"
            placeholder="Write a review"
            value={text}
          />
          <Button type="submit" isDisabled={btnDisabled}>
            Send
          </Button>
        </div>
        {message && <div className="message">{message}</div>}
      </form>
    </Card>
  );
}

export default FeedbackForm;
```

FeedbackItem.jsx

```jsx
import { FaTimes } from "react-icons/fa";
import PropTypes from "prop-types";
import Card from "./shared/Card";
import { useContext } from "react";
import FeedbackContext from "../context/FeedbackContext";

function FeedbackItem({ item }) {
  const { deleteFeedback } = useContext(FeedbackContext);
    // we did same thing like FeedbackForm and change onClick event function to deleteFeedback

  return (
    <Card>
      <div className="num-display">{item.rating}</div>
      <button onClick={() => deleteFeedback(item.id)} className="close">
        <FaTimes color="purple" />
      </button>
      <div className="text-display">{item.text}</div>
    </Card>
  );
}

FeedbackItem.propTypes = {
  item: PropTypes.object.isRequired,
};

export default FeedbackItem;
```

* EditFeedback Process

FeedbackContext.js

```jsx
import { createContext, useState } from "react";
import { v4 as uuidv4 } from "uuid";

const FeedbackContext = createContext();

export const FeedbackProvider = ({ children }) => {
  const [feedback, setFeedback] = useState([
    {
      id: 1,
      text: "This is feedback item 1",
      rating: 10,
    }
```

```
    {
      id: 2,
      text: "This is feedback item 2",
      rating: 9,
    },
    {
      id: 3,
      text: "This is feedback item 3",
      rating: 7,
    },
  ]);
  const [feedbackEdit, setFeedbackEdit] = useState({
    item: {},
    edit: false, // We set false in first hand.
  }); // We set another useState for edit.


  // Delete Feedback
  const deleteFeedback = (id) => {
    if (window.confirm("Are you sure you want to delete?")) {
      setFeedback(feedback.filter((item) => item.id !== id));
    }
  };

  // Set item to be update

  const editFeedback = (item) => {
    setFeedbackEdit({ // We create a function when we click edit key will be true.
      item,
      edit: true,
    });
  };

  // Add Feedback
  const addFeedback = (newFeedback) => {
    newFeedback.id = uuidv4();
    setFeedback([newFeedback, ...feedback]);
  };

  return (
    <FeedbackContext.Provider
      value={{
        feedback,
        deleteFeedback,
        addFeedback,
        editFeedback, // We send it to FeedbackItem to use as onClick Event
      }}
    >
      {children}
    </FeedbackContext.Provider>
  );
};

export default FeedbackContext;


// Another Component

 FeedbackItem.jsx

 import { FaTimes, FaEdit } from "react-icons/fa";
import PropTypes from "prop-types";
import Card from "./shared/Card";
import { useContext } from "react";
import FeedbackContext from "../context/FeedbackContext";

function FeedbackItem({ item }) {
  const { deleteFeedback, editFeedback } = useContext(FeedbackContext);
    // First we add editFeedback here

  return (
    <Card>
      <div className="num-display">{item.rating}</div>
      <button onClick={() => deleteFeedback(item.id)} className="close">
        <FaTimes color="purple" />
      </button>
      <button className="edit" onClick={() => editFeedback(item)}> {// from FeedbackContext
        <FaEdit color="purple" />
      </button>
      <div className="text-display">{item.text}</div>
```

```
    </Card>
  );
}

FeedbackItem.propTypes = {
  item: PropTypes.object.isRequired,
};

export default FeedbackItem;
```

```
/* MULTIPLE COMPONENT */


FeedbackContext.js

import { createContext, useState } from "react";
import { v4 as uuidv4 } from "uuid";

const FeedbackContext = createContext();

export const FeedbackProvider = ({ children }) => {
  const [feedback, setFeedback] = useState([
    {
      id: 1,
      text: "This is feedback item 1",
      rating: 10,
    },
    {
      id: 2,
      text: "This is feedback item 2",
      rating: 9,
    },
    {
      id: 3,
      text: "This is feedback item 3",
      rating: 7,
    },
  ]);
  const [feedbackEdit, setFeedbackEdit] = useState({ // Default
    item: {},
    edit: false,
  });
  // Delete Feedback
  const deleteFeedback = (id) => {
    if (window.confirm("Are you sure you want to delete?")) {
      setFeedback(feedback.filter((item) => item.id !== id));
    }
  };

  // Set item to be update

  const editFeedback = (item) => {
    setFeedbackEdit({
      item,
      edit: true,
    });
  };

  // Add Feedback
  const addFeedback = (newFeedback) => {
    newFeedback.id = uuidv4();
    setFeedback([newFeedback, ...feedback]);
  };

  // Update Feedback Item
  const updateFeedback = (id, updItem) => { // For change we sent to params.
    setFeedback(
      feedback.map((item) => (item.id === id ? { ...item, ...updItem } : item))
    );
```

```
    );
  };

  return (
    <FeedbackContext.Provider
      value={{
        feedback,
        deleteFeedback,
        addFeedback,
        editFeedback,
        feedbackEdit, // We send this as value to FeedbackForm cause we'll change input's value.
        updateFeedback, // This is for change to input's value
      }}
    >
      {children}
    </FeedbackContext.Provider>
  );
};

export default FeedbackContext;
```

FeedbackItem.jsx

```
import { FaTimes, FaEdit } from "react-icons/fa";
import PropTypes from "prop-types";
import Card from "./shared/Card";
import { useContext } from "react";
import FeedbackContext from "../context/FeedbackContext";

function FeedbackItem({ item }) {
  const { deleteFeedback, editFeedback } = useContext(FeedbackContext);

    // WE import deleteFeedback and EditFeedback
  return (
    <Card>
      <div className="num-display">{item.rating}</div>
      <button onClick={() => deleteFeedback(item.id)} className="close">
        <FaTimes color="purple" />
      </button>
      <button className="edit" onClick={() => editFeedback(item)}>
          {// When we click this we set FeedbackEdit.edit as true.
        <FaEdit color="purple" />
      </button>
      <div className="text-display">{item.text}</div>
    </Card>
  );
}

FeedbackItem.propTypes = {
  item: PropTypes.object.isRequired,
};

export default FeedbackItem;
```

FeedbackForm.jsx

```
import { useState, useContext, useEffect } from "react";
import Card from "./shared/Card";
import Button from "./shared/Button";
import RatingSelect from "./RatingSelect";
import FeedbackContext from "../context/FeedbackContext";

function FeedbackForm() {
  const [text, setText] = useState("");
  const [rating, setRating] = useState(10);
  const [btnDisabled, setBtnDisabled] = useState(true);
  const [message, setMessage] = useState("");

  const { addFeedback, feedbackEdit, updateFeedback } =
    useContext(FeedbackContext);

  useEffect(() => { // We add feedbackEdit and updateFeedback above.
    if (feedbackEdit.edit === true) { // Here we check the condition
      setBtnDisabled(false); // We activate to button
      setText(feedbackEdit.item.text); // We catch text send as value to input
      setRating(feedbackEdit.item.rating);// We catch rating send as value to ratingarea also we
have to use, useEffect at RatingSelect.jsx
    }
```

```jsx
  }, [feedbackEdit]); // This is for the catch array when we click edit icon.It comes from
FeedbackItem.jsx
  const handleTextChange = (e) => {
    if (text === "") {
      setBtnDisabled(true);
      setMessage(null);
    } else if (text !== "" && text.trim().length <= 10) {
      setMessage("Text must be at least 10 characters long");
      setBtnDisabled(true);
    } else {
      setMessage(null);
      setBtnDisabled(false);
    }
    setText(e.target.value);
  };

  const handleSubmit = (e) => {
    e.preventDefault();
    if (text.trim().length > 10) {
      const newFeedback = {
        text,
        rating,
      };

      if (feedbackEdit.edit === true) { // In here we use updateFeedback for change Feedback values.
If our condition is true then it will be added like newFeedback but only change to values
        updateFeedback(feedbackEdit.item.id, newFeedback);
      } else {
        addFeedback(newFeedback);
      }

      setText("");
    }
  };

  return (
    <Card>
      <form onSubmit={handleSubmit}>
        <h2>How would you rate your service with us?</h2>
        <RatingSelect select={(rating) => setRating(rating)} />
        <div className="input-group">
          <input
            onChange={handleTextChange}
            type="text"
            placeholder="Write a review"
            value={text}
          />
          <Button type="submit" isDisabled={btnDisabled}>
            Send
          </Button>
        </div>
        {message && <div className="message">{message}</div>}
      </form>
    </Card>
  );
}

export default FeedbackForm;
```

RatingSelect.jsx

```jsx
import { useState, useEffect, useContext } from "react";
import FeedbackContext from "../context/FeedbackContext";
function RatingSelect({ select }) {
  const [selected, setSelected] = useState(10);

  const { feedbackEdit } = useContext(FeedbackContext);

  useEffect(() => { /// We check the feedbackEdit.edit if its true, now we selectedRating updating
for ratingarea
    if (feedbackEdit.edit === true) {
      setSelected(feedbackEdit.item.rating);
    }
  }, [feedbackEdit]);

  const handleChange = (e) => {
    setSelected(+e.currentTarget.value);
    select(+e.currentTarget.value);
  };

  return ( // Nothing wrong below, color problem about carbon.no
    <ul className="rating">
      {Array.from({ length: 10 }, (_, i) => (
        <li key={`rating-${i + 1}`}>
          <input
            type="radio"
            id={`num${i + 1}`}
            name="rating"
            value={i + 1}
            onChange={handleChange}
            checked={selected === i + 1}
          />
          <label htmlFor={`num${i + 1}`}>{i + 1}</label>
        </li>
      ))}
    </ul>
  );
}

export default RatingSelect;
```

// There is also typora format of codes

RatingSelect.jsx

```jsx
import { useState, useEffect, useContext } from "react";
import FeedbackContext from "../context/FeedbackContext";
function RatingSelect({ select }) {
  const [selected, setSelected] = useState(10);
```

```
  const { feedbackEdit } = useContext(FeedbackContext);

  useEffect(() => { // We check the feedbackEdit.edit if its true, now
we selectedRating updating for ratingarea
    if (feedbackEdit.edit === true) {
      setSelected(feedbackEdit.item.rating);
    }
  }, [feedbackEdit]);

  const handleChange = (e) => {
    setSelected(+e.currentTarget.value);
    select(+e.currentTarget.value);
  };

  return (
    <ul className="rating">
      {Array.from({ length: 10 }, (_, i) => (
        <li key={`rating-${i + 1}`}>
          <input
            type="radio"
            id={`num${i + 1}`}
            name="rating"
            value={i + 1}
            onChange={handleChange}
            checked={selected === i + 1}
          />
          <label htmlFor={`num${i + 1}`}>{i + 1}</label>
        </li>
      ))}
    </ul>
  );
}

export default RatingSelect;
```

We clicked edit button,text went to input area and rating selected like we selected before.