

```

import React, { useContext, useEffect, useState } from "react";
import { AiOutlineClose } from "react-icons/ai";
import { BsLinkedin } from "react-icons/bs";
import { GoMarkGithub } from "react-icons/go";
import { MdDeleteForever } from "react-icons/md";
import { RiEdit2Fill } from "react-icons/ri";
import { TfiTwitterAlt } from "react-icons/tfi";
import { VscAdd } from "react-icons/vsc";
import { AppContext } from "../context/AppContext";
import {
  getUserSkills,
  removeUserSkills,
  saveUserSkills,
  updateUserDetails,
} from "../proxies/backend_api";

const Profile = () => {
  const { user, setShowAlert, setUser, skills, setSkills } =
    useContext(AppContext);

  const [addSkill, setAddSkill] = useState("");

  const [newSkills, setNewSkills] = useState([]);

  const [removedSkills, setRemovedSkills] = useState([]);

  const [isEditingEnabled, setIsEditingEnabled] = useState(false);

  const [userInfo, setUserInfo] = useState({
    name: "",
    phone_number: "",
  });

  const handleUserInfoChange = ({ target: { name, value } }) => {
    setUserInfo((prev) => ({ ...prev, [name]: value }));
  };

  const changeSkills = () => {
    if (
      addSkill !== "" &&
      !skills.find((item) => item.toLowerCase() === addSkill.toLowerCase())
    ) {
      setNewSkills((prev) => [...prev, addSkill.trim()]);
      setSkills((prev) => [...prev, addSkill.trim()]);
    }
    setAddSkill("");
  };

  const removeSkills = (skill_name) => {
    setRemovedSkills((prev) => [...prev, skill_name]);
    setSkills((prev) => prev.filter((item) => item !== skill_name));
    setNewSkills((prev) => prev.filter((item) => item !== skill_name));
  };

  const updateSkills = async () => {

```

```

let skillsAdded = false,
    skillsRemoved = false;
if (newSkills.length !== 0) {
    skillsAdded = await saveUserSkills(newSkills, user.token);
}
if (removeSkills.length !== 0) {
    skillsRemoved = await removeUserSkills(removedSkills, user.token);
}
if (skillsAdded || skillsRemoved) {
    setShowAlert({
        type: "success",
        message: "Profile updated!",
        duration: 3000,
    });
}
setNewSkills([]);
setRemovedSkills([]);
};

const updateUserInfo = async () => {
    const data = await updateUserDetails(userInfo, user.token);
    if (data) {
        setUser((prev) => {
            prev = { ...prev, name: data.name, phone_number: data.phone_number };
            localStorage.setItem("user", JSON.stringify(prev));
            return prev;
        });
        setShowAlert({
            type: "success",
            message: "Profile updated!",
            duration: 3000,
        });
    }
    setIsEditingEnabled(false);
};

useEffect(() => {
    if (user) {
        (async () => {
            let data = await getUserSkills(user?.token);
            if (data) setSkills(data);
        })();
        setUserInfo({
            name: user.name,
            phone_number: user.phone_number,
        });
    }
}, [user]);

return (
    <div className="my-5 mx-10">
        <div className="border-2 border-blue-100 w-full h-fit rounded-xl p-5 flex flex-col gap-3">
            <div className="flex justify-between w-full min-h-[25vh]">
                <div className="flex flex-col justify-between">
                    <h1 className="md:text-2xl text-xl font-medium flex items-center gap-4">

```

```

Your Profile{ " " }
<button>
  {isEditingEnabled ? (
    <AiOutlineClose
      color="#ff8977"
      onClick={() => setIsEditingEnabled(!isEditingEnabled)}
    />
  ) : (
    <RiEdit2Fill
      color="#4506cb"
      onClick={() => setIsEditingEnabled(!isEditingEnabled)}
    />
  )}
</button>
</h1>
<div className="flex flex-col gap-3">
  {isEditingEnabled ? (
    <>
      <input
        name="name"
        value={userInfo.name}
        className="input input-bordered w-full input-xs p-3 text-lg input-primary"
        type="text"
        placeholder="name"
        onChange={handleUserInfoChange}
      />
      <input
        disabled
        value={user?.email}
        className="input input-bordered w-full input-xs p-3 text-lg input-primary"
        type="text"
        placeholder="name"
      />
      <input
        name="phone_number"
        value={userInfo.phone_number}
        className="input input-bordered w-full input-xs p-3 text-lg input-primary"
        type="number"
        placeholder="phone number"
        onChange={handleUserInfoChange}
      />
      <button
        className="btn btn-xs btn-outline btn-primary"
        onClick={updateUserInfo}
      >
        Update
      </button>
    </>
  ) : (
    <>
      <h2 className="md:text-2xl xl:text-2xl sm:text-xl">
        {user?.name}
      </h2>
      <p className="md:text-xl sm:text-md text-gray-700">
        {user?.email}
      </p>
    </>
  )}

```

```

    </p>
    <span className="text-gray-700">{user?.phone_number}</span>
  </>
  )}
</div>
</div>
<div className="flex flex-col justify-end w-fit gap-4">
  
  { /* <button className="btn btn-outline btn-active btn-sm">
    change
  </button> */ }
</div>
</div>
<div className="divider my-2"></div>
<div className="flex flex-col">
  <div className="flex justify-between gap-2 flex-col">
    <h4 className="text-xl">Skills</h4>
    <form
      className="flex gap-5 items-center"
      onSubmit={(e) => e.preventDefault()}
    >
      <input
        autoComplete="off"
        value={addSkill}
        type="text"
        name="addSkill"
        placeholder="add addSkill"
        onChange={(e) => setAddSkill(e.target.value)}
        className="input input-bordered w-full input-primary max-w-xl input-sm"
      />

      <button
        className="hover:rotate-90 transition-all"
        onClick={changeSkills}
      >
        <VscAdd size={20} />
      </button>
    </form>
    <ul className="flex gap-2 flex-wrap">
      {skills?.map((addSkill, ind) => (
        <li
          className="bg-indigo-100 rounded p-2 flex gap-2 items-center"
          key={ind}
        >
          {addSkill}
          <MdDeleteForever
            color="#ff8977"
            onClick={() => removeSkills(addSkill)}
            size={20}
          />
        </li>
      )
    )}

```

```

    ))}
  </ul>
  <button
    className="btn btn-sm w-fit btn-primary"
    type="button"
    onClick={updateSkills}
  >
    Save
  </button>
</div>
<div className="divider my-2"></div>
<div className="flex justify-between gap-2 flex-col">
  <h4 className="text-xl">Resume/Portfolio</h4>
  <div className="flex gap-5">
    <input
      className="input input-bordered w-full input-primary max-w-xl input-sm"
      type="text"
      placeholder="paste the link"
    />
    <button className="btn btn-primary btn-sm">update</button>
  </div>
</div>
<div className="divider my-2"></div>
<div className="flex gap-2 flex-col">
  <h3 className="text-xl">Socials</h3>
  <div className="flex flex-col gap-2">
    <div className="flex gap-5 items-center">
      <GoMarkGithub size={20} />
      <input
        type="text"
        placeholder="paste the link"
        className="border-2 border-gray-300 rounded-md px-3 my-1 max-w-md"
      />
    </div>
    <div className="flex gap-5 items-center">
      <BsLinkedin size={20} />
      <input
        type="text"
        placeholder="paste the link"
        className="border-2 border-gray-300 rounded-md px-3 my-1 max-w-md"
      />
    </div>
    <div className="flex gap-5 items-center">
      <TfiTwitterAlt size={20} />
      <input
        type="text"
        placeholder="paste the link"
        className="border-2 border-gray-300 rounded-md px-3 my-1 max-w-md"
      />
    </div>
    <button className="btn btn-primary btn-sm max-w-fit">save</button>
  </div>
</div>
</div>
</div>

```

```
    </div>  
  );  
};  
  
export default Profile;
```