import React from "react";

import '../App.css';

import TextField from "@mui/material/TextField";

const About = () => {

  return (

    <div style={{paddingBottom:"20px"}}>

      <TextField

        required

        id="outlined-required"

        label="About You"

        defaultValue="Lorem ipsum dolor sit amet consectetur. Nisi viverra pulvinar cursus morbi aliquet gravida tincidunt lobortis non. Sed ut leo magna pulvinar odio amet in. Lorem ipsum dolor sit amet consectetur. Nisi viverra pulvinar cursus morbi."

        multiline

        rows={4}

        variant="outlined"

        fullWidth

      />

    </div>

  );

};

export default About;

import React from "react";

const KeyAndValue = ({ props }) => {

  return (

    <div style={{ padding: "10px"}}>

      <div style={{ fontWeight: "bold" }}>{props.title}</div>

      <div>{props.value}</div>

    </div>

  );

};

export default KeyAndValue;

import React from "react";

const NameWithIcon = ({ name, image }) => {

  return (

    <div>

      <div style={{ display: "flex", gap: "10px", alignItems: "center" }}>

        <div>{name}</div>

        <div>

          <img

            src={image}

            style={{ width: "30px", height: "30px", borderRadius: "50%" }}

          />

        </div>

      </div>

    </div>

  );

};

export default NameWithIcon;

import React, { useEffect, useState } from "react";

import { useFormik } from "formik";

import TextField from "@mui/material/TextField";

import MenuItem from "@mui/material/MenuItem";

import { Button, Checkbox, FormControlLabel, IconButton } from "@mui/material";

import View from "./View";

import basicShema from "../schemas";

import "../App.css";

import { useNavigate, useLocation } from "react-router-dom";

import About from "./About";

const defaultValues = {

  firstName: "Steven",

  lastName: "Quadros",

  gender: "Male",

  dob: "",

  email: "",

  phoneNumber: "",

  nationality: "",

  locality: "",

  permanentAddress: "",

  linkedin: "https://www.linkedin.com/in/username123",

  pinCode: "",

  district: "",

  state: "",

  github: "https://github.com/user123",

  languages: [

    {

      language: "",

      proficiency: "",

    },

    { language: "", proficiency: "" },

  ],

};

const PersonalDetailsForm = ({ onSave }) => {

  const [isSaved, setIsSaved] = useState(false);

  const navigate = useNavigate();

  const location = useLocation();

  const [initialValues, setInitialValues] = useState(defaultValues);

  const languageOptions = ["English", "Hindi", "Tamil", "French"];

  const proficiencyOptions = ["Beginner", "Intermediate", "Advanced", "Fluent"];

  useEffect(() => {

    if (location.state?.data) {

      setInitialValues(location.state.data);

    }

  }, [location.state]);

  const {

    values,

    handleBlur,

    touched,

    handleChange,

    errors,

    handleSubmit,

    resetForm,

    setFieldValue,

  } = useFormik({

    initialValues: initialValues,

    enableReinitialize: true,

    validationSchema: basicShema,

    onSubmit: (values) => {

      onSave(values);

      navigate("/view-details", { state: { data: values } });

    },

  });

  const handleDiscard = () => {

    resetForm({

      values: defaultValues,

    });

    setIsSaved(false);

  };

  const handleChangeInternal = (e) => {

    handleChange(e);

    setIsSaved(false);

  };

   const addLanguage = () => {

     setFieldValue("languages", [

       ...values.languages,

       { language: "", proficiency: "" },

     ]);

   };

   const removeLanguage = (index) => {

     const updatedLanguages = values.languages.filter((\_, i) => i !== index);

     setFieldValue("languages", updatedLanguages);

   };

  return (

    <div className="app-container">

      <div

        style={{

          border: "2px solid rgb(143, 190, 226)",

          padding: "40px",

          borderRadius: "15px",

        }}

      >

        {isSaved && <View data={[values]} />}

        <h3 style={{ paddingBottom: "30px" }}>Personal Details</h3>

        <About />

        <form onSubmit={handleSubmit} autoComplete="off">

          <div className="form-container">

            <div className="form-column">

              <TextField

                required

                id="firstName"

                name="firstName"

                label="First Name"

                value={values.firstName}

                onChange={handleChangeInternal}

                onBlur={handleBlur}

                error={touched.firstName && Boolean(errors.firstName)}

                helperText={touched.firstName && errors.firstName}

                variant="outlined"

                fullWidth

              />

              <TextField

                select

                required

                id="gender"

                name="gender"

                label="Gender"

                value={values.gender}

                onChange={handleChangeInternal}

                onBlur={handleBlur}

                error={touched.gender && Boolean(errors.gender)}

                helperText={touched.gender && errors.gender}

                variant="outlined"

                fullWidth

              >

                <MenuItem value="Male">Male</MenuItem>

                <MenuItem value="Female">Female</MenuItem>

                <MenuItem value="Other">Other</MenuItem>

              </TextField>

              <TextField

                required

                id="email"

                name="email"

                label="Email"

                type="email"

                value={values.email}

                onChange={handleChangeInternal}

                onBlur={handleBlur}

                error={touched.email && Boolean(errors.email)}

                helperText={touched.email && errors.email}

                variant="outlined"

                fullWidth

              />

              <TextField

                select

                required

                id="nationality"

                name="nationality"

                label="Nationality"

                value={values.nationality}

                onChange={handleChangeInternal}

                onBlur={handleBlur}

                error={touched.nationality && Boolean(errors.nationality)}

                helperText={touched.nationality && errors.nationality}

                variant="outlined"

                fullWidth

              >

                <MenuItem value="Indian">Indian</MenuItem>

                <MenuItem value="American">American</MenuItem>

                <MenuItem value="British">British</MenuItem>

              </TextField>

              <TextField

                select

                required

                id="state"

                name="state"

                label="State"

                value={values.state}

                onChange={handleChangeInternal}

                onBlur={handleBlur}

                error={touched.state && Boolean(errors.state)}

                helperText={touched.state && errors.state}

                variant="outlined"

                fullWidth

              >

                <MenuItem value="Tamil Nadu">Tamil Nadu</MenuItem>

                <MenuItem value="Karnataka">Karnataka</MenuItem>

                <MenuItem value="Kerala">Kerala</MenuItem>

              </TextField>

              <TextField

                select

                required

                id="locality"

                name="locality"

                label="Locality"

                value={values.locality}

                onChange={handleChangeInternal}

                onBlur={handleBlur}

                error={touched.locality && Boolean(errors.locality)}

                helperText={touched.locality && errors.locality}

                variant="outlined"

                fullWidth

              >

                <MenuItem value="Adyar">Adyar</MenuItem>

                <MenuItem value="Velachery">Velachery</MenuItem>

                <MenuItem value="T. Nagar">T. Nagar</MenuItem>

              </TextField>

              <TextField

                id="linkedin"

                name="linkedin"

                label="LinkedIn Profile URL"

                value={values.linkedin}

                onChange={handleChangeInternal}

                onBlur={handleBlur}

                error={touched.linkedin && Boolean(errors.linkedin)}

                helperText={touched.linkedin && errors.linkedin}

                variant="outlined"

                fullWidth

              />

            </div>

            <div className="form-column">

              <TextField

                required

                id="lastName"

                name="lastName"

                label="Last Name"

                value={values.lastName}

                onChange={handleChangeInternal}

                onBlur={handleBlur}

                error={touched.lastName && Boolean(errors.lastName)}

                helperText={touched.lastName && errors.lastName}

                variant="outlined"

                fullWidth

              />

              <TextField

                required

                id="dob"

                name="dob"

                label="Date of Birth"

                type="date"

                InputLabelProps={{ shrink: true }}

                value={values.dob}

                onChange={handleChangeInternal}

                onBlur={handleBlur}

                error={touched.dob && Boolean(errors.dob)}

                helperText={touched.dob && errors.dob}

                variant="outlined"

                fullWidth

              />

              <TextField

                required

                id="phoneNumber"

                name="phoneNumber"

                label="Phone Number"

                value={values.phoneNumber}

                onChange={handleChangeInternal}

                onBlur={handleBlur}

                error={touched.phoneNumber && Boolean(errors.phoneNumber)}

                helperText={touched.phoneNumber && errors.phoneNumber}

                variant="outlined"

                fullWidth

              />

              <TextField

                required

                id="pinCode"

                name="pinCode"

                label="Pin Code"

                value={values.pinCode}

                onChange={handleChangeInternal}

                onBlur={handleBlur}

                error={touched.pinCode && Boolean(errors.pinCode)}

                helperText={touched.pinCode && errors.pinCode}

                variant="outlined"

                fullWidth

              />

              <TextField

                select

                required

                id="district"

                name="district"

                label="District"

                value={values.district}

                onChange={handleChangeInternal}

                onBlur={handleBlur}

                error={touched.district && Boolean(errors.district)}

                helperText={touched.district && errors.district}

                variant="outlined"

                fullWidth

              >

                <MenuItem value="Chennai">Chennai</MenuItem>

                <MenuItem value="Coimbatore">Coimbatore</MenuItem>

                <MenuItem value="Madurai">Madurai</MenuItem>

              </TextField>

              <TextField

                required

                id="permanentAddress"

                name="permanentAddress"

                label="Permanent Address"

                value={values.permanentAddress}

                onChange={handleChangeInternal}

                onBlur={handleBlur}

                error={

                  touched.permanentAddress && Boolean(errors.permanentAddress)

                }

                helperText={touched.permanentAddress && errors.permanentAddress}

                variant="outlined"

                fullWidth

              />

              <TextField

                id="github"

                name="github"

                label="GitHub Profile URL"

                value={values.github}

                onChange={handleChangeInternal}

                onBlur={handleBlur}

                error={touched.github && Boolean(errors.github)}

                helperText={touched.github && errors.github}

                variant="outlined"

                fullWidth

              />

            </div>

          </div>

          {values.languages.map((lang, index) => (

            <div key={index} style={{ marginBottom: "20px" }}>

              <div

                style={{

                  display: "flex",

                  justifyContent: "space-between",

                  gap: "20px",

                }}

              >

                <TextField

                  select

                  id={`languages.${index}.language`}

                  name={`languages.${index}.language`}

                  label="Language"

                  value={lang.language}

                  onChange={handleChange}

                  onBlur={handleBlur}

                  error={

                    touched.languages?.[index]?.language &&

                    Boolean(errors.languages?.[index]?.language)

                  }

                  helperText={

                    touched.languages?.[index]?.language &&

                    errors.languages?.[index]?.language

                  }

                  variant="outlined"

                  style={{ flex: 1 }}

                >

                  {languageOptions.map((option) => (

                    <MenuItem key={option} value={option}>

                      {option}

                    </MenuItem>

                  ))}

                </TextField>

                <TextField

                  select

                  id={`languages.${index}.proficiency`}

                  name={`languages.${index}.proficiency`}

                  label="Proficiency"

                  value={lang.proficiency}

                  onChange={handleChange}

                  onBlur={handleBlur}

                  error={

                    touched.languages?.[index]?.proficiency &&

                    Boolean(errors.languages?.[index]?.proficiency)

                  }

                  helperText={

                    touched.languages?.[index]?.proficiency &&

                    errors.languages?.[index]?.proficiency

                  }

                  variant="outlined"

                  style={{ flex: 1 }}

                >

                  {proficiencyOptions.map((option) => (

                    <MenuItem key={option} value={option}>

                      {option}

                    </MenuItem>

                  ))}

                </TextField>

              </div>

              <div

                style={{

                  display: "flex",

                  justifyContent: "space-between",

                  marginTop: "10px",

                }}

              >

                <div style={{ display: "flex", gap: "20px" }}>

                  <FormControlLabel

                    control={

                      <Checkbox

                        id={`languages.${index}.read`}

                        name={`languages.${index}.read`}

                        checked={lang.read}

                        onChange={handleChange}

                      />

                    }

                    label="Read"

                  />

                  <FormControlLabel

                    control={

                      <Checkbox

                        id={`languages.${index}.write`}

                        name={`languages.${index}.write`}

                        checked={lang.write}

                        onChange={handleChange}

                      />

                    }

                    label="Write"

                  />

                  <FormControlLabel

                    control={

                      <Checkbox

                        id={`languages.${index}.speak`}

                        name={`languages.${index}.speak`}

                        checked={lang.speak}

                        onChange={handleChange}

                      />

                    }

                    label="Speak"

                  />

                </div>

                <IconButton

                  onClick={() => removeLanguage(index)}

                  disabled={values.languages.length <= 1}

                  style={{

                    color: values.languages.length <= 1 ? "gray" : "blue",

                    fontWeight: "bold",

                    fontSize: "15px",

                  }}

                >

                  Delete

                </IconButton>

              </div>

            </div>

          ))}

          <div>

            <Button

              onClick={addLanguage}

              variant="contained"

              disabled={values.languages.length >= 10}

            >

              Add Language

            </Button>

          </div>

          <div style={{ paddingTop: "20px" }}>

            <Button variant="contained" type="submit">

              Save Changes

            </Button>

            <Button variant="text" onClick={handleDiscard}>

              Discard Changes

            </Button>

          </div>

        </form>

      </div>

    </div>

  );

};

export default PersonalDetailsForm;

import React from "react";

import { useLocation } from "react-router-dom";

import KeyAndValue from "./KeyAndValue";

import { useNavigate } from "react-router-dom";

import About from "./About";

const View = () => {

  const location = useLocation();

  const navigate = useNavigate();

  const data = location.state?.data || {};

  const handleEditClick = () => {

    navigate("/", { state: { data: data } });

  };

  return (

    <div className="app-container">

      <div

        style={{

          border: "2px solid rgb(143, 190, 226)",

          padding: "40px",

          borderRadius: "15px",

        }}

      >

        <div

          style={{

            display: "flex",

            justifyContent: "space-between",

            alignItems: "center",

          }}

        >

          <div>

            <h3>Personal Details</h3>

            <div>20% Complete</div>

          </div>

          <div

            style={{

              color: "#0000FF",

              padding: "20px",

              fontWeight: "bold",

              cursor: "pointer",

            }}

          >

            <div onClick={handleEditClick}>Edit Details</div>

          </div>

        </div>

        <div style={{ padding: "30px 0px" }}>

          <About />

        </div>

        <div style={{ display: "flex", width: "100%" }}>

          <div

            style={{

              display: "flex",

              flexDirection: "column",

              flex: 1,

              padding: "10px",

            }}

          >

            <KeyAndValue

              props={{

                title: "First Name",

                value: data.firstName || "Pending.....",

              }}

            />

            <KeyAndValue

              props={{ title: "Gender", value: data.gender || "Pending....." }}

            />

            <KeyAndValue

              props={{ title: "Email ID", value: data.email || "Pending....." }}

            />

            <KeyAndValue

              props={{

                title: "Nationality",

                value: data.nationality || "Pending.....",

              }}

            />

            <KeyAndValue

              props={{ title: "State", value: data.state || "Pending....." }}

            />

            <KeyAndValue

              props={{

                title: "Locality",

                value: data.locality || "Pending.....",

              }}

            />

            <KeyAndValue

              props={{

                title: "Linkedin Profile URL",

                value: data.linkedin || "Pending.....",

              }}

            />

          </div>

          <div

            style={{

              display: "flex",

              flexDirection: "column",

              flex: 1,

              padding: "10px",

            }}

          >

            <KeyAndValue

              props={{

                title: "Last Name",

                value: data.lastName || "Pending.....",

              }}

            />

            <KeyAndValue

              props={{

                title: "Date of Birth",

                value: data.dob || "Pending.....",

              }}

            />

            <KeyAndValue

              props={{

                title: "Phone number",

                value: data.phoneNumber || "Pending.....",

              }}

            />

            <KeyAndValue

              props={{

                title: "Pin code",

                value: data.pinCode || "Pending.....",

              }}

            />

            <KeyAndValue

              props={{

                title: "District",

                value: data.district || "Pending.....",

              }}

            />

            <KeyAndValue

              props={{

                title: "Permanent Address",

                value: data.address || "Pending.....",

              }}

            />

            <KeyAndValue

              props={{

                title: "Github profile link",

                value: data.github || "Pending.....",

              }}

            />

          </div>

        </div>

        <h4>Language</h4>

        <div

          style={{ display: "flex", flexDirection: "column", width: "100%" }}

        >

          <div

            style={{

              display: "flex",

              fontWeight: "bold",

              borderBottom: "1px solid #ccc",

              paddingBottom: "5px",

            }}

          >

            <div style={{ flex: 2 }}>Language</div>{" "}

            <div style={{ flex: 2 }}>Proficiency</div>

            <div style={{ flex: 1, textAlign: "center" }}>Read</div>

            <div style={{ flex: 1, textAlign: "center" }}>Write</div>

            <div style={{ flex: 1, textAlign: "center" }}>Speak</div>

          </div>

          {data.languages?.slice(0, 10).map((lang, index) => (

            <div

              key={index}

              style={{

                display: "flex",

                padding: "5px 0",

              }}

            >

              <div style={{ flex: 2 }}>{lang.language || "Pending....."}</div>

              <div style={{ flex: 2 }}>

                {lang.proficiency || "Pending....."}

              </div>

              <div style={{ flex: 1, textAlign: "center" }}>

                {lang.read ? (

                  <span style={{ color: "green", fontSize: "20px" }}>✔</span>

                ) : (

                  <span style={{ color: "red", fontSize: "20px" }}>❌</span>

                )}

              </div>

              <div style={{ flex: 1, textAlign: "center" }}>

                {lang.write ? (

                  <span style={{ color: "green", fontSize: "20px" }}>✔</span>

                ) : (

                  <span style={{ color: "red", fontSize: "20px" }}>❌</span>

                )}

              </div>

              <div style={{ flex: 1, textAlign: "center" }}>

                {lang.speak ? (

                  <span style={{ color: "green", fontSize: "20px" }}>✔</span>

                ) : (

                  <span style={{ color: "red", fontSize: "20px" }}>❌</span>

                )}

              </div>

            </div>

          ))}

        </div>

      </div>

    </div>

  );

};

export default View;

import React, { useEffect, useState } from "react";

import KeyAndValue from "./KeyAndValue";

const ViewDetails = () => {

  const [personalData, setPersonalData] = useState(null);

  useEffect(() => {

    const fetchPersonalData = async () => {

      try {

        const response = await fetch(

          "https://api.postalpincode.in/pincode/600017"

        );

        const data = await response.json();

        console.log(data);

        if (data && data[0]?.PostOffice) {

          setPersonalData(data[0].PostOffice);

        } else {

          setPersonalData([]);

        }

      } catch (error) {

        console.error("Error fetching data:", error);

        setPersonalData([]);

      }

    };

    fetchPersonalData();

  }, []);

  return (

    <div>

      <h1>Personal Details</h1>

      {personalData?.map((item, index) => (

        <div key={index} style={{ display: "flex" }}>

          <div style={{ display: "flex", flexDirection: "column" }}>

            <KeyAndValue props={{ title: "First Name", value: item.Name }} />

            <KeyAndValue

              props={{ title: "Gender", value: item.Gender || "N/A" }}

            />

            <KeyAndValue

              props={{ title: "Email ID", value: item.Email || "N/A" }}

            />

            <KeyAndValue

              props={{ title: "Nationality", value: item.Country || "N/A" }}

            />

            <KeyAndValue

              props={{ title: "City", value: item.District || "N/A" }}

            />

            <KeyAndValue

              props={{

                title: "Linkedin Profile URL",

                value: item.Linkedin || "N/A",

              }}

            />

          </div>

          <div style={{ display: "flex", flexDirection: "column" }}>

            <KeyAndValue

              props={{ title: "Last Name", value: item.LastName || "N/A" }}

            />

            <KeyAndValue

              props={{ title: "Date of Birth", value: item.Dob || "N/A" }}

            />

            <KeyAndValue

              props={{

                title: "Phone Number",

                value: item.PhoneNumber || "N/A",

              }}

            />

            <KeyAndValue

              props={{ title: "State", value: item.State || "N/A" }}

            />

            <KeyAndValue

              props={{ title: "Address", value: item.Address || "N/A" }}

            />

            <KeyAndValue

              props={{ title: "Pin Code", value: item.Pincode || "N/A" }}

            />

          </div>

        </div>

      ))}

    </div>

  );

};

export default ViewDetails;

import \* as yup from "yup";

const basicShema = yup.object().shape({

  firstName: yup

    .string()

    .min(2, "First name must be at least 2 characters")

    .max(15, "First name cannot exceed 30 characters")

    .required("This field is required"),

  lastName: yup

    .string()

    .min(2, "Last name must be at least 2 characters")

    .max(15, "Last name cannot exceed 30 characters")

    .required("This field is required"),

  gender: yup.string().required("This field is required"),

  dob: yup.string().required("This field is required"),

  email: yup

    .string()

    .email("Please enter a valid email address")

    .required("This field is required"),

  phoneNumber: yup

    .string()

    .matches(/^\d{10}$/, "Please enter a valid 10-digit phone number")

    .required("This field is required"),

  nationality: yup.string().required("This field is required"),

  pinCode: yup

    .string()

    .length(6, "Pin code must be exactly 6 digits")

    .required("This field is required"),

  state: yup.string().required("This field is required"),

  district: yup.string().required("This field is required"),

  locality: yup.string().required("This field is required"),

  permanentAddress: yup.string().required("This field is required"),

  linkedin: yup.string()

    .url("Please enter a valid LinkedIn URL")

    .matches(

      /^(https?:\/\/)?(www\.)?linkedin\.com\/.\*$/,

      "Please enter a valid LinkedIn URL"

    ),

  github: yup.string()

    .url("Please enter a valid GitHub URL")

    .matches(

      /^(https?:\/\/)?(www\.)?github\.com\/.\*$/,

      "Please enter a valid GitHub URL"

    ),

  language: yup.string(),

  proficiency: yup.string(),

  skills: yup.string(),

});

export default basicShema;

.app-container {

  padding: 40px;

}

.error{

  color:red;

}

.form-container {

  display: grid;

  grid-template-columns: 1fr 1fr;

  gap: 20px;

  width: 100%;

  padding-bottom: 20px;

}

.form-column {

  display: flex;

  flex-direction: column;

  gap: 20px;

}

import "./App.css";

import PersonalDetailsForm from "./Component/PersonalDetailsForm";

import View from "./Component/View";

import { BrowserRouter, Routes, Route } from "react-router-dom";

function App() {

  const handleSave = (data) => {

    console.log("Saved data:", data);

  };

  return (

    <BrowserRouter>

      <div>

        <Routes>

          <Route path="/" element={<PersonalDetailsForm onSave={handleSave} />} />

          <Route path="/view-details" element={<View />} />

        </Routes>

      </div>

    </BrowserRouter>

  );

}

export default App;

output:









