// import './UpdateStudentDetailsFormCss.css';

import { FooterStyle } from "../../StyleCss/FooterStyle";

import React from "react";

import { armyUnits, statuses } from "../constants";

import { EventType } from "@testing-library/react";

import axios from "axios";

import { KeyObject } from "crypto";

import { useEffect, useState } from "react";

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

import { studentDTO, Students1 } from "../../../Types/ModelsStudents";

import { variables } from "../../../Variables";

import { homeStatus } from "../constants";

import DeleteForeverOutlinedIcon from "@mui/icons-material/DeleteForeverOutlined";

import "./UpdateStudentCss.css";

import {

  Box,

  FormControl,

  FormControlLabel,

  FormLabel,

  MenuItem,

  Radio,

  RadioGroup,

  Select,

  Step,

  StepLabel,

  Stepper,

  TextField,

} from "@mui/material";

import { DatePicker, LocalizationProvider } from "@mui/x-date-pickers";

import { AdapterDayjs } from "@mui/x-date-pickers/AdapterDayjs";

const UpdateStudentDetailsForm = ({

  myStudent,

  onNextClick,

  setMyStudent,

  activeStep,

  setActiveStep,

  setShowFilesForm

}: {

  myStudent: studentDTO;

  onNextClick: () => void;

  setMyStudent: (student: studentDTO) => void;

  activeStep: any;

  setActiveStep: any;

  setShowFilesForm: any;

}) => {

  const navigate = useNavigate();

  const [errors, setErrors] = useState<any>({});

  const steps = ["", ""];

  const [file, setFile] = useState(null);

  const params = useParams();

  const [value, setValue] = React.useState({});

  useEffect(() => {

    getStudent();

  }, []);

  const getStudent = async () => {

    const res = await axios.get(

      variables.API\_URL +

        `Student/GetStudentDetailsByParams?FirstName=default&LastName=default&StatusId=-1&Id=${params.id}`

    );

    setMyStudent(res.data[0]);

  };

  const UploadImage = (data: FormData) => {

    axios

      .post(

        variables.API\_URL +

          `Student/UploadImage?studentId=${myStudent.studentId}`,

        data,

        {

          headers: {

            "content-type": "multipart/form-data",

          },

        }

      )

      .then((res) => {

        console.log(res.statusText);

      });

  };

  const handleChangeInt = (e: any, atrr: keyof Students1) => {

    setMyStudent({ ...myStudent, [atrr]: parseInt(e.target.value) });

  };

  const handleChangeString = (e: any, atrr: keyof Students1) => {

    setMyStudent({ ...myStudent, [atrr]: e.target.value });

  };

  const handleChangeDate = (e: any, atrr: keyof Students1) => {

    setMyStudent({ ...myStudent, [atrr]: new Date(e.target.value) });

  };

  const radioHandler1 = (event: React.ChangeEvent<HTMLInputElement>) => {

    myStudent.wasInSeminar = Boolean(event.target.value);

    console.log(event.target.value);

  };

  const radioHandler = (

    event: React.ChangeEvent<HTMLInputElement>,

    atrr: keyof Students1

  ) => {

    console.log("=====", event.target.value);

    setMyStudent({ ...myStudent, [atrr]: Boolean(event.target.value) });

  };

  const handleFileChangePicture = (event: any) => {

    setFile(event.target.files[0]);

    setMyStudent({ ...myStudent, picture: event.target.files[0].name });

  };

  const handleFileChangeIdFront = (event: any) => {

    setFile(event.target.files[0]);

    setMyStudent({ ...myStudent, idCardFront: event.target.files[0].name });

  };

  const handleFileChangeIdBack = (event: any) => {

    setFile(event.target.files[0]);

    setMyStudent({ ...myStudent, idCardBack: event.target.files[0].name });

  };

  const handleFileChangeIdAttachment = (event: any) => {

    setFile(event.target.files[0]);

    setMyStudent({

      ...myStudent,

      idCardAttachment: event.target.files[0].name,

    });

  };

  const refreshPage = () => {

    window.location.reload();

  };

  const handleSubmit = async (event: any) => {

    event.preventDefault();

    const data = new FormData();

    data.append("userfile", file != null ? file : "");

    await UploadImage(data);

    // refreshPage()

  };

  const addNewStudent = () => {

    navigate("/AddNewStudent");

  };

  const sendALinkForRegistration = () => {

    navigate("/SendALinkForRegistration");

  };

  const getSeminars = () => {

    navigate("/ViewSeminars");

  };

  const addSeminar = () => {

    navigate("/AddSeminar");

  };

  const excel = () => {

    navigate("/ExcelTrying");

  };

  const main = () => {

    navigate("/");

  };

  const handleChange = (event: React.ChangeEvent<HTMLInputElement>) => {

    setValue((event.target as HTMLInputElement).value);

  };

  const handleChange1 = (event: React.ChangeEvent<HTMLInputElement>) => {

    setValue((event.target as HTMLInputElement).value);

  };

  const changeHomeStatus = (event: any) => {

    handleChangeString(event, "homeStatus");

  };

  return (

    <>

      <div>

        <Box id="stepper" sx={{ width: "15%" }}>

          <Stepper activeStep={activeStep} alternativeLabel>

            {steps.map((label, index) => (

              <Step key={index}>

                <StepLabel onClick={() => setActiveStep(index)}>

                {label}

                </StepLabel >

              </Step>

            ))}

          </Stepper>

        </Box>

        <h1 id="titleDetails"> עדכון פרטים אישיים</h1>

        <h1 id="titleDetails1">פרטי התקשרות</h1>

        <h1 id="titleDetails2">כתובת מגורים</h1>

        <form>

          <FormControl>

            <FormLabel id="demo-radio-buttons-group-label">

              <legend className="containerChavruta">?מעוניין בחברותא</legend>

            </FormLabel>

            <RadioGroup

              value={myStudent.wantChavruta}

              className="containerChavrutaUpdate"

              onChange={(event, value) => {

                console.log(myStudent.wantChavruta.toString());

                if (value === "true")

                  return setMyStudent({ ...myStudent, wantChavruta: true });

                return setMyStudent({ ...myStudent, wantChavruta: false });

              }}

            >

              <FormControlLabel value={"true"} control={<Radio />} label="כן" />

              <FormControlLabel

                value={"false"}

                control={<Radio />}

                label="לא"

              />

            </RadioGroup>

          </FormControl>

          <FormControl className="labelUpdateSeminar">

            <FormLabel id="demo-radio-buttons-group-label">

              השתתף ביום עיון

            </FormLabel>

            <RadioGroup

              value={myStudent.wasInSeminar.toString()}

              className="containerSeminarUpdate"

              onChange={(event, value) => {

                console.log(myStudent.wasInSeminar);

                if (value === "true")

                  return setMyStudent({ ...myStudent, wasInSeminar: true });

                return setMyStudent({ ...myStudent, wasInSeminar: false });

              }}

            >

              <FormControlLabel value={"true"} control={<Radio />} label="כן" />

              <FormControlLabel

                value={"false"}

                control={<Radio />}

                label="לא"

              />

            </RadioGroup>

          </FormControl>

          <br></br>

          <Box

            className="status1"

            component="form"

            sx={{

              "& .MuiTextField-root": { m: 1, width: "25ch" },

            }}

            noValidate

            autoComplete="off"

          >

            <div>

              <TextField

                id="outlined-select-currency"

                select

                label="סטטוס"

                value={myStudent?.statusId}

                // helperText="סמן מהו הסטטוס שלך"

                onChange={(e) => {

                  handleChangeInt(e, "statusId");

                }}

              >

                {statuses.map((option) => (

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

                    {option.label}

                  </MenuItem>

                ))}

              </TextField>

            </div>

          </Box>

          <TextField

            value={myStudent?.firstName|| ""}

            className="firstName1"

            id="outlined-basic"

            label="שם פרטי "

            variant="outlined"

            type={"string"}

            required

            onChange={(e) => {

              handleChangeString(e, "firstName");

            }}

          />

          <TextField

            value={myStudent?.studentId}

            className="studentId1"

            id="outlined-basic"

            label="תז"

            variant="outlined"

            type={"string"}

            required

            onChange={(e) => {

              handleChangeString(e, "studentId");

            }}

          />

          {/\* <TextField

              value={myStudent?.birthdate}

              className="birthdate1"

              id="outlined-basic"

              label="תאריך לידה"

              variant="outlined"

              type={"string"}

              required

              onChange={(e) => {

                handleChangeDate(e, "birthdate");

              }}

            /> \*/}

          {/\* <LocalizationProvider

            value={myStudent?.birthdate}

            id="date1"

            className="birthDate11"

            required

            variant="outlined"

            type={"string"}

            onChange={(e:any ) => {

                handleChangeDate(e, "birthdate");

              }}

            dateAdapter={AdapterDayjs}

            sx={{

              "& .MuiInputBase-input-MuiOutlinedInput-input": { top: "300px" },

            }}

          >

            <DatePicker />

          </LocalizationProvider> \*/}

          {/\* <input

            className="birthdateStudent1"

            type={"date"}

            onChange={(e: any) => {

              handleChangeDate(e, "birthdate");

            }}

          ></input> \*/}

          <TextField

            value={myStudent?.birthdate}

            className="birthdateStudent1"

            id="outlined-basic"

            label="תאריך"

            variant="outlined"

            type={"string"}

            required

            onChange={(e) => {

              handleChangeDate(e, "birthdate");

            }}

          />

          <TextField

            value={myStudent?.lastName}

            className="lastName1"

            id="outlined-basic"

            label="שם משפחה "

            variant="outlined"

            type={"string"}

            required

            onChange={(e) => {

              handleChangeString(e, "lastName");

            }}

          />

          <br></br>

          <TextField

            value={myStudent?.adress}

            className="adress1"

            id="outlined-basic"

            label="כתובת "

            variant="outlined"

            type={"string"}

            required

            onChange={(e) => {

              handleChangeString(e, "adress");

            }}

          />

          <br></br>

          <TextField

            value={myStudent?.zipCode}

            className="zipCode1"

            id="outlined-basic"

            label="מיקוד "

            variant="outlined"

            type={"string"}

            required

            onChange={(e) => {

              handleChangeString(e, "zipCode");

            }}

          />

          <TextField

            value={myStudent?.phone}

            className="phone1"

            id="outlined-basic"

            label="פלאפון"

            variant="outlined"

            type={"string"}

            required

            onChange={(e) => {

              handleChangeString(e, "phone");

            }}

          />

          <TextField

            value={myStudent?.email}

            className="email1"

            id="outlined-basic"

            label="מייל"

            variant="outlined"

            type={"string"}

            required

            onChange={(e) => {

              handleChangeString(e, "email");

            }}

          />

          <TextField

            value={myStudent?.school}

            className="school1"

            id="outlined-basic"

            label="בית ספר תיכון "

            variant="outlined"

            type={"string"}

            required

            onChange={(e) => {

              handleChangeString(e, "school");

            }}

          />

          <TextField

            value={myStudent?.profession}

            className="profession1"

            id="outlined-basic"

            label=" מקצוע"

            variant="outlined"

            type={"string"}

            required

            onChange={(e) => {

              handleChangeString(e, "profession");

            }}

          />

          <br></br>

          <br></br>

          <TextField

            className="army1"

            id="outlined-select-currency"

            select

            label="יחידת שירות בצבא"

            value={myStudent?.armiUnit}

            // helperText="סמן מהו הסטטוס שלך"

            onChange={(e) => {

              handleChangeString(e, "armiUnit");

            }}

          >

            {armyUnits.map((option) => (

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

                {option.label}

              </MenuItem>

            ))}

          </TextField>

          <br></br>

          <TextField

            value={myStudent?.homeStatus}

            className="homeStatus1"

            id="outlined-basic"

            label="מצב הבית"

            variant="outlined"

            type={"string"}

            required

            onChange={(e) => {

              handleChangeString(e, "homeStatus");

            }}

          />

          {/\* <Box

              className="homeStatus"

              component="form"

              sx={{

                "& .MuiTextField-root": { m: 1, width: "25ch" },

              }}

              noValidate

              autoComplete="off"

            >

              <div>

                <TextField

                  id="outlined-select-currency"

                  select

                  label="מ"

                  value={student?.homeStatus}

                  // helperText="סמן מהו הסטטוס שלך"

                  onChange={(e) => {

                    handleChangeString(e, "homeStatus");

                  }}

                >

                  {homeStatus.map((option) => (

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

                      {option.label}

                    </MenuItem>

                  ))}

                </TextField>

              </div>

            </Box>

            <br></br> \*/}

          <TextField

            value={myStudent?.fatherName}

            className="fatherName1"

            id="outlined-basic"

            label="שם האב "

            variant="outlined"

            type={"string"}

            required

            onChange={(e) => {

              handleChangeString(e, "fatherName");

            }}

          />

          <TextField

            value={myStudent?.motherName}

            className="motherName1"

            id="outlined-basic"

            label="שם האם "

            variant="outlined"

            type={"string"}

            required

            onChange={(e) => {

              handleChangeString(e, "motherName");

            }}

          />

        </form>

        <button

          id="nextButton"

          onClick={() => {

            const hasErrors = Object.values(errors).some((e) => !!e);

            hasErrors && alert("הנתונים לא חוקיים");

            !hasErrors && onNextClick();

          }}

          // disabled={}

        >

          הבא

        </button>

        <button id="garbageButton" onClick={refreshPage}>

          {

            <DeleteForeverOutlinedIcon

              style={{ width: "35px", height: "35px", color: "white" }}

            />

          }

        </button>

        {<FooterStyle></FooterStyle>}

      </div>

    </>

  );

};

export default UpdateStudentDetailsForm;