**Number of hours / number of hours per week = number of weeks**

**Number of hours / hours per lesson = number of lessons**

**Find on which days the flow holds classes.**

**Loop through the days of the week and schedule a class for each day that is marked as a class day. When the counter reaches the number of days stop.**

**This process starts once the flow has a start date.**

**Use flow\_id to get spec\_id from the flow table. Get all courses for the spec from the spec\_courses table.**

**user**

1.    \_id

2.    user\_name

3.    role Lecturer Student Admin

2.    name + last

3.    email

4.    phone

4.    hash

5.    salt

**course**

1.    \_id AI NN

2.    name

3.    hours

4.    lecturer\_id

5.    file

**speciality**

1.    \_id AI NN

2.    name

**speciality\_course**

1.    speciality\_id

2.    course\_id

**flow**

1.    \_id AI NN

2.    speciality\_id

3.    start\_date

4.    a 1/0

5.    b

6.    c

7.    d

8.    e

9.    f

**flow\_student**

1.    flow\_id

2.    student\_id

**time\_table**

1.    date: Date

2.    course\_id

**lessons**

1.    date: Date

2.    course\_id

3.    student\_id

4.    present 1/0

**calendar**

1.    date Date

2.    holiday 1/0

Att

1. student\_id

2. Jan 1st

3.

Pages

1.

Nav Bar with routes

2.

Login as card

3.

Students / Lecturer (table)

Student form new/update

4.

Course Form new/update

Course Table

4.1.

Speciality-course (choose course for speciality with option to add course or remove course) Table to Table

5.

Flow Form + (table sort on dates)

5.1.

flow student (choose student for flow with option to add student or remove student) Table to Table

6.

Time table - calendar (moment) or ready calendar

7.

Lesson - from calendar - Table of students for current lesson

https://github.com/olegkorzun/jwt-nodejs.git

**user**

1. \_id
2. user\_name
3. role
4. hash
5. salt

**course**

1. \_id AI NN
2. name
3. hours
4. lecturer\_id
5. file

**student**

1. \_id (user\_id)
2. name
3. email
4. phone

**lecturer**

1. \_id (user\_id)
2. name
3. email
4. phone

**speciality**

1. \_id AI NN
2. name

**speciality\_course**

1. speciality\_id
2. course\_id

**flow**

1. \_id AI NN
2. speciality\_id
3. start\_date
4. a 1/0
5. b
6. c
7. d
8. e
9. f

**flow\_student**

1. flow\_id
2. student\_id

**lessons**

1. date
2. course\_id
3. student\_id
4. present 1/0

**calendar**

1. date Date
2. holiday 1/0

import React from "react";

import "../index.css"

interface Props {

}

interface State {

}

class registration extends React.Component<Props, State> {

    constructor(props: Props) {

        super(props);

    }

    render() {

        return (

            <div className="card card-body bg-light m-4">

                <form>

                    <h3>Registration</h3>

                    <div className="form-group">

                        <label>First Name</label>

                        <input type="text" className="form-control" placeholder="enter first name" />

                    </div>

                    <div className="form-group">

                        <label>Last Name</label>

                        <input type="email" className="form-control" placeholder="enter last name" />

                    </div>

                    <div className="form-group">

                        <label>Username</label>

                        <input type="email" className="form-control" placeholder="enter username" />

                    </div>

                    <div className="form-group">

                        <label>Password</label>

                        <input type="password" className="form-control" placeholder="enter password" />

                    </div>

                    <div className="form-group">

                        <label>Select Role</label>

                        <select className="select-css">

                            <option value="val1">teacher</option>

                            <option value="val2">student</option>

                            <option value="val3">administrator</option>

                        </select>

                    </div>

                    <div className="form-group">

                        <label>Phone Number</label>

                        <input type="text" className="form-control" placeholder="enter phone number" />

                    </div>

                    <div className="form-group">

                        <label>ID</label>

                        <input type="text" className="form-control" placeholder="enter ID" />

                    </div>

                    <button type="submit" className="btn btn-primary btn-block">Submit</button>

                    <button type="button" className="btn btn-outline-primary mr-3">clear form</button>

                </form>

            </div>

        );

    }

}

export default registration

// import React, { Component } from 'react'

// import "../../index.css"

// import AdminNav from "./AdminNav";

// type RegData = {

//     status: string;

//     data: Array<Reg>;

// };

// type Reg = {

//     user\_id: number;

//     personal\_id: number;

//     first\_name: string;

//     last\_name: string;

//     role: string;

//     hash: string;

//     email: string;

//     phone: string

// };

// interface Props { }

// interface State {

//     registrations: Array<Reg>;

//     user\_id: number;

//     personal\_id: number;

//     first\_name: string;

//     last\_name: string;

//     role: string;

//     hash: string;

//     email: string;

//     phone: string

// }

// export class Registration extends Component<Props, State> {

//     constructor(props: Props) {

//         super(props);

//         this.state = {

//             registrations: [],

//             user\_id: 0,

//             personal\_id: 0,

//             first\_name: "",

//             last\_name: "",

//             role: "",

//             hash: "",

//             email: "",

//             phone: ""

//         };

//     }

//     handlePersonalIdChange = (e: React.ChangeEvent<any>) => {

//         this.setState({

//             personal\_id: e.target.value

//         })

//     };

//     handleFirstNameChange = (e: React.ChangeEvent<HTMLInputElement>) => {

//         this.setState({

//             first\_name: e.target.value

//         })

//     };

//     handleLastNameChange = (e: React.ChangeEvent<HTMLInputElement>) => {

//         this.setState({

//             last\_name: e.target.value

//         })

//     };

//     handleRoleChange = (e: React.ChangeEvent<HTMLSelectElement>) => {

//         this.setState({

//             role: e.target.value

//         })

//     };

//     handleHashChange = (e: React.ChangeEvent<HTMLInputElement>) => {

//         this.setState({

//             hash: e.target.value

//         })

//     };

//     handleEmailChange = (e: React.ChangeEvent<HTMLInputElement>) => {

//         this.setState({

//             email: e.target.value

//         })

//     };

//     handlePhoneChange = (e: React.ChangeEvent<HTMLInputElement>) => {

//         this.setState({

//             phone: e.target.value

//         })

//     };

//     addUser = () => {

//         // let h = (document.getElementById("roleSelect") as HTMLSelectElement).value

//         // this.setState({

//         //     role: h

//         // })

//         if (this.state.first\_name === ""){

//             return alert("first name must be completed");

//         }

//         if (this.state.personal\_id === 0){

//             return alert("id must be completed");

//         }

//         if (this.state.hash === ""){

//             return alert("hash must be completed");

//         }

//         if (this.state.email === ""){

//             return alert("email must be completed");

//         }

//         if (this.state.role === ""){

//             return alert("role must be completed");

//         }

//         // || this.state.last\_name === "" ||

//         //     this.state.personal\_id <= 0 || this.state.role === "" ||

//         //     this.state.hash === "" ||

//         //     this.state.email === "") {

//         //     return alert("all fields must be completed");

//         // }

//         fetch('/Registration/addUser', {

//             method: 'POST',

//             headers: {

//                 'Content-Type': 'application/json'

//             },

//             body: JSON.stringify({

//                 personal\_id: this.state.personal\_id,

//                 first\_name: this.state.first\_name,

//                 last\_name: this.state.last\_name,

//                 role: this.state.role,

//                 hash: this.state.hash,

//                 email: this.state.email

//             })

//         })

//             .then(res => {

//                 if (res.ok) {

//                     alert("user added successfully!");

//                     this.clearForm();

//                 }

//             })

//             .catch(error => console.log(error));

//     };

//     // deleteUser = () => {

//     //     fetch('/Registration/deleteUser', {

//     //         method: 'DELETE',

//     //         headers: {

//     //             'Content-Type': 'application/json'

//     //         },

//     //         body: JSON.stringify({ personal\_id: this.state.personal\_id })

//     //     })

//     //         .then(res => {

//     //             if (res.ok) {

//     //                 this.clearForm();

//     //             }

//     //         })

//     //         .catch(error => console.log(error));

//     // };

//     clearForm = () => {

//         this.setState({

//             registrations: [],

//             personal\_id: 0,

//             first\_name: "",

//             last\_name: "",

//             role: "",

//             hash: "",

//             email: "",

//             phone: ""

//           });

//         };

//     // updateUser = () => {

//     //     fetch('/Registration/updateUser', {

//     //         method: 'PATCH',

//     //         headers: {

//     //             'Content-Type': 'application/json'

//     //         },

//     //         body: JSON.stringify({

//     //             user\_id: this.state.user\_id,

//     //             personal\_id: this.state.personal\_id,

//     //             first\_name: this.state.first\_name,

//     //             last\_name: this.state.last\_name,

//     //             role: this.state.role,

//     //             hash: this.state.hash,

//     //             email: this.state.email

//     //         })

//     //     })

//     //         .then(res => {

//     //             if (res.ok) {

//     //                 this.clearForm();

//     //             }

//     //         })

//     //         .catch(error => console.log(error));

//     // };

//     // showCourseName = (reg: Reg) => {

//     //     this.setState({

//     //         user\_id: reg.user\_id,

//     //         personal\_id: reg.personal\_id,

//     //         first\_name: reg.first\_name,

//     //         last\_name: reg.last\_name,

//     //         role: reg.role,

//     //         hash: reg.hash,

//     //         email: reg.email,

//     //         registrations: this.state.registrations

//     //     });

//     // };

//     render() {

//         return (

//             <div>

//                 <AdminNav/>

//             <div className="card card-body bg-light m-4">

//                 <form name="formid">

//                     <h3>Registration</h3>

//                     <div className="form-group">

//                         <label htmlFor="nameField">First Name</label>

//                         <input type="text"

//                         className="form-control"

//                         placeholder="enter first name"

//                         value={this.state.first\_name}

//                         onChange={this.handleFirstNameChange} />

//                     </div>

//                     <div className="form-group">

//                         <label htmlFor="nameField">Last Name</label>

//                         <input type="text"

//                         className="form-control"

//                         placeholder="enter last name"

//                         value={this.state.last\_name}

//                         onChange={this.handleLastNameChange} />

//                     </div>

//                     <div className="form-group">

//                         <label htmlFor="nameField">Username</label>

//                         <input type="text"

//                         className="form-control"

//                         placeholder="enter username"

//                         value={this.state.email}

//                         onChange={this.handleEmailChange} />

//                     </div>

//                     <div className="form-group">

//                         <label htmlFor="nameField">Password</label>

//                         <input type="password"

//                         className="form-control"

//                         placeholder="enter password"

//                         value={this.state.hash}

//                         onChange={this.handleHashChange} />

//                     </div>

//                     <div className="form-group">

//                         <label htmlFor="nameField">Select Role</label>

//                         <select className="select-css" onChange={this.handleRoleChange}>

//                             <option value="">click to select role</option>

//                             <option value="lecturer">lecturer</option>

//                             <option value="student">student</option>

//                             <option value="admin">administrator</option>

//                         </select>

//                     </div>

//                     <div className="form-group">

//                         <label htmlFor="nameField">Phone Number</label>

//                         <input type="text"

//                         className="form-control"

//                         placeholder="enter phone number"

//                         value={this.state.phone}

//                         onChange={this.handlePhoneChange}  />

//                     </div>

//                     <div className="form-group">

//                         <label htmlFor="nameField">ID</label>

//                         <input type="text"

//                         className="form-control"

//                         placeholder="enter ID"

//                         value={this.state.personal\_id}

//                         onChange={this.handlePersonalIdChange} />

//                     </div>

//                     <button type="submit" className="btn btn-primary btn-block"

//                     onClick={this.addUser}>Add User</button>

//                     {/\* <button type="submit" className="btn btn-primary btn-block"

//                     onClick={this.updateUser}>Update User</button>

//                     <button type="submit" className="btn btn-primary btn-block"

//                     onClick={this.deleteUser}>Delete User</button>

//                     <button type="button"

//                     className="btn btn-outline-primary mr-3"

//                     onClick={this.clearForm}>clear form</button> \*/}

//                 </form>

//             </div>

//             </div>

//         );

//     }

// }

// export default Registration

// var express = require('express');

// var myRegistration = require('../controllers/myRegistration');

// var router = express.Router();

// router.get('/', function(req, res, next) {

//   res.send('Registration Page');

// });

// router.post('/addUser', myRegistration.addUser);

// module.exports = router;

// var getConnection = require('./database');

// var mysql = require("mysql2");

// exports.addUser = function(req, res, next){

//     var first\_name = req.body.first\_name;

//     var last\_name = req.body.last\_name;

//     var email = req.body.email;

//     var role = req.body.role;

//     var hash = req.body.hash;

//     var phone = req.body.phone;

//     console.log("==========", first\_name);

//     console.log("==========", last\_name);

//     console.log("==========", role);

//         if (!first\_name || first\_name.length === 0 ||

//         !last\_name || last\_name.length === 0 ||

//         !personal\_id || personal\_id.length === 0 ||

//         !email || email.length === 0 ||

//         !role || role.length === 0 ||

//         !hash || hash.length === 0){

//             console.log("ERROR: Task name is empty");

//         res.json({

//             "status": "error",

//             "msg": "ERROR:information fields are empty"

//         });

//             return;

//     }

//     getConnection(function(err, conn){

//         if(err){

//             throw err;

//         }

//         console.log("connected successfully");

//         var sql = "INSERT INTO `user` (`first\_name`, `last\_name`, `role`, `hash`, " +

//         "`email`, `phone`) "+

//         "VALUES (" + mysql.escape(first\_name) + ", " + mysql.escape(last\_name) +

//         ", " + mysql.escape(role) + ", " + mysql.escape(hash) + ", " + mysql.escape(email)+

//         ", " + mysql.escape(phone) +");";

//         conn.query(sql, function(err, userData){

//             if(err){

//                 throw err;

//             }

//             res.json({

//                 "status": "ok",

//                 data: userData

//             });

//             conn.release();

//         });

//     });

// };

// registration.js in controllers

// var getConnection = require('./database');

// var mysql = require("mysql2");

// // exports.createUsersTable  = function(req, res, next){

// //     console.log("createTable");

// //     getConnection(function(err, conn){

// //         if(err){

// //             throw err;

// //         }

// //         console.log("connected successfully");

// //     var  sql  =  `

// //         CREATE TABLE IF NOT EXISTS users (

// //         id integer PRIMARY KEY,

// //         name varchar(255) NULL,

// //         email varchar(190) UNIQUE,

// //         password varchar(190))`;

// //         conn.query(sql, function(err, result){

// //             if(err){

// //                 throw err;

// //             }

// //             res.send(result);

// //         });

// //     });

// // };

// // exports.deleteUser = function(req, res, next){

// //     var user\_id = req.body.user\_id;

// //     console.log("==========", user\_id);

// //         if (!user\_id){

// //         console.log("ERROR: user\_id is empty");

// //         res.json({

// //             "status": "error",

// //             "msg": "ERROR:user\_id is empty"

// //         });

// //             return;

// //     }

// //     getConnection(function(err, conn){

// //         if(err){

// //             throw err;

// //         }

// //         console.log("connected successfully");

// //         var sql = "DELETE FROM `user` WHERE user\_id=" +

// //         mysql.escape(user\_id) + ";";

// //         conn.query(sql, function(err, userDeleted){

// //             if(err){

// //                 throw err;

// //             }

// //             res.json({

// //                 "status": "ok",

// //                 data: userDeleted

// //             });

// //             conn.release();

// //         });

// //     });

// // };

// exports.addUser = function(req, res, next){

//     var first\_name = req.body.first\_name;

//     var last\_name = req.body.last\_name;

//     var email = req.body.email;

//     var role = req.body.role;

//     var hash = req.body.hash;

//     var phone = req.body.phone;

//     console.log("==========", first\_name);

//     console.log("==========", last\_name);

//     console.log("==========", role);

//         if (!first\_name || first\_name.length === 0 ||

//         !last\_name || last\_name.length === 0 ||

//         !personal\_id || personal\_id.length === 0 ||

//         !email || email.length === 0 ||

//         !role || role.length === 0 ||

//         !hash || hash.length === 0){

//             console.log("ERROR: Task name is empty");

//         res.json({

//             "status": "error",

//             "msg": "ERROR:information fields are empty"

//         });

//             return;

//     }

//     getConnection(function(err, conn){

//         if(err){

//             throw err;

//         }

//         console.log("connected successfully");

//         var sql = "INSERT INTO `user` (`first\_name`, `last\_name`, `role`, `hash`, " +

//         "`email`, `phone`) "+

//         "VALUES (" + mysql.escape(first\_name) + ", " + mysql.escape(last\_name) +

//         ", " + mysql.escape(role) + ", " + mysql.escape(hash) + ", " + mysql.escape(email)+

//         ", " + mysql.escape(phone) +");";

//         conn.query(sql, function(err, userData){

//             if(err){

//                 throw err;

//             }

//             res.json({

//                 "status": "ok",

//                 data: userData

//             });

//             conn.release();

//         });

//     });

// };

// // exports.updateTask = function(req, res, next){

// //     var taskId = req.body.taskId;

// //     var taskName = req.body.taskName;

// //     var lastName = req.body.lastName;

// //     var email = req.body.email;

// //     // todo: add validation check

// //         if (!taskId){

// //         console.log("ERROR: Task id is empty");

// //         res.json({

// //             "status": "error",

// //             "msg": "ERROR:task id is empty"

// //         });

// //             return;

// //     }

// //     getConnection(function(err, conn){

// //         if(err){

// //             throw err;

// //         }

// //         console.log("connected successfully");

// //         var sql = "UPDATE `students` SET fname=" + mysql.escape(taskName) +

// //             ", lname=" + mysql.escape(lastName) +

// //             ", email=" + mysql.escape(email) +

// //             " WHERE student\_id=" +

// //         mysql.escape(taskId) + ";";

// //         conn.query(sql, function(err, tasks){

// //             if(err){

// //                 throw err;

// //             }

// //             res.json({

// //                 "status": "ok",

// //                 data: tasks

// //             });

// //             conn.release();

// //         });

// //     });

// // };

// ";

// registration in routes

// var express = require('express');

// var registration = require('../controllers/registration');

// var router = express.Router();

// router.get('/', function(req, res, next) {

//   res.send('Registration Page');

// });

// //router.get('/add-task', task.addTask);

// //router.get('/createTaskTbl', tasks.createTable);

// router.post('/addUser', registration.addUser);

// // router.delete('/deleteUser', registration.deleteUser);

// // router.patch('/updateUser', registration.updateUser);

// module.exports = router;