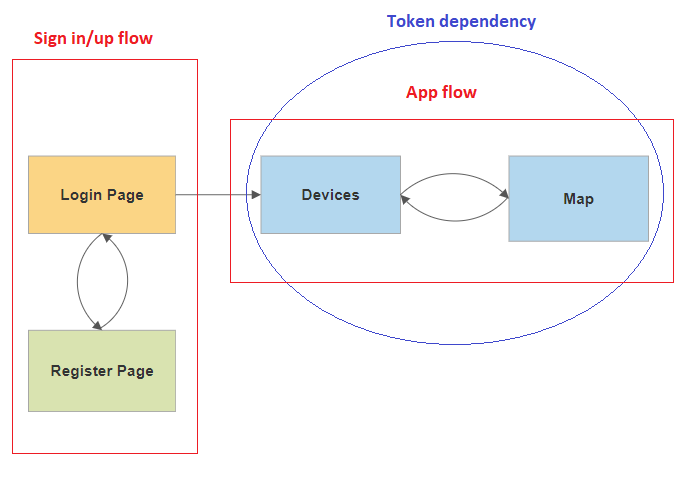
## **Aplicatie GPS**

**Motiv si uitilitate tema**

Motivul alegerii acestei teme a fost eficienta si utilitatea pe care o au sistemele GPS in viata de zi cu zi, ajutand la o mai buna orientare in spatiu si la o mai buna organizare a traficului precum si estimarea timpului si constientizarea pericolelor existente intr-o zona mare de supraveghere.

**Structura aplicatiei**

****

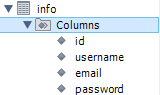
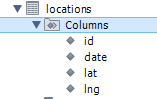
**Instructiuni aplicatie**

* Register Page : aici va inregistrati cu datele dumneavoastra, urmand ca ulterior sa puteti accesa aplicatia fara probleme.
* Login Page : aici o sa introduceti datele cu care v-ati inregistrat, iar dupa apasarea butonului Login, un token cu id-ul dumneavoastra sa va fie asignat si sa puteti folosi aplicatia.
* Device Page : interfata web este constituita din doua campuri text care contin niste date: prima data constituie limita inferioara, iar cea de a doua limita superioara, urmand ca la apasarea butonului “Get Positions” sa obtineti toate pozitiile inregistrate intre datele de mai sus afisate intr-o lista precum si un buton aditional care o sa va ajute sa eliberati display-ul oricand doriti. Cel de-al doilea buton (“ Show on map “) va ajuta sa puteti sa accesati Map Page
* Map Page : contine o harta care va avea plasate mai multe markere fixe care constituie pozitia fiecarui user care a dorit ca pozitia lui sa fie publica. Contine doua butoane pentru plasare si pentru stergerea marker-ului user-ului actual. Un buton de revenire la pagina device si un buton de logout.

**Cod sursa**

***Backend/ API + DB***

Avem doua baze de date: una creata pentru primul flow, iar cealalta pentru cel de-al doilea.

Doua servere create pe domeniile: localhost port 8000 si respectiv port 9000.

*server1.js*

const express = require('express');

const bodyparser = require('body-parser');

const registeringRoutes = require('./routes/registeringRoutes');

var cors = require('cors');

var app = express();

app.use(bodyparser.json());

app.use(bodyparser.urlencoded({ extended: false }));

app.use(cors());

app.use("/",registeringRoutes);

app.listen(8000, () => {

    console.log("Server is listening on port 8000");

});

*server2.js*

const express = require('express');

const bodyparser = require('body-parser');

const locationsRoutes = require('./routes/locationsRoutes');

var cors = require('cors');

var app = express();

app.use(bodyparser.json());

app.use(bodyparser.urlencoded({ extended: false }));

app.use(cors());

app.use("/",locationsRoutes);

app.listen(9000, () => {

    console.log("Server is listening on port 9000");

});

CRUD-ul aferent pentru ambele baze de date.

*registeringRoutes.js*

const express = require('express');

const Router = express.Router();

const sqlConnection = require('../sqlConnection');

const idGenerator = require('../functions/idGenerator');

Router.get("/", (req, res) => {

    sqlConnection.query("SELECT \* from info", (err, rows) => {

        if(!err){

            res.send(rows);

        }

        else{

            console.log(err);

        }

    })

})

Router.post('/insert',(req, res) => {

      const data = {

                        id: idGenerator(),

                        username: req.query.username,

                        email: req.query.email,

                        password: req.query.password

                    };

      const sql = "INSERT INTO info SET ?";

      sqlConnection.query(sql, data,(err) => {

        if(err) throw err;

        res.redirect('/');

      });

  });

Router.post('/update',(req, res) => {

    const sql = "UPDATE info SET username='"+req.query.username+"', email='"+req.query.email+"', password='"+req.query.password+"' WHERE id='"+req.query.id+"'";

      sqlConnection.query(sql, (err) => {

        if(err) throw err;

        res.redirect('/');

      });

    });

Router.post('/delete',(req, res) => {

      const sql = "DELETE FROM info WHERE id='"+req.query.id+"'";

      sqlConnection.query(sql, (err) => {

        if(err) throw err;

          res.redirect('/');

      });

    });

module.exports = Router;

*locationsRoutes.js*

const express = require('express');

const Router = express.Router();

const sqlConnection = require('../sqlConnection');

Router.get("/", (req, res) => {

    sqlConnection.query("SELECT \* from locations", (err, rows) => {

        if(!err){

            res.send(rows);

        }

        else{

            console.log(err);

        }

    })

})

Router.post('/insert',(req, res) => {

      const data = {

                        id: req.query.id,

                        date: req.query.date,

                        lat: req.query.lat,

                        lng: req.query.lng

                    };

      const sql = "INSERT INTO locations SET ?";

      sqlConnection.query(sql, data,(err) => {

        if(err) throw err;

        res.redirect('/');

      });

  });

Router.post('/update',(req, res) => {

    const sql = "UPDATE locations SET lat='"+req.query.lat+"', lng='"+req.query.lng+"' WHERE id='"+req.query.id+"'";

      sqlConnection.query(sql, (err) => {

        if(err) throw err;

        res.redirect('/');

      });

    });

Router.post('/delete',(req, res) => {

      const sql = "DELETE FROM locations WHERE id='"+req.query.id+"'";

      sqlConnection.query(sql, (err) => {

        if(err) throw err;

          res.redirect('/');

      });

    });

module.exports = Router;

Functia de generare a unui id.

idGenerator.js

const idGenerator = () => '\_' + Math.random().toString(36).substr(2, 9);

module.exports = idGenerator;

***Frontend/ Client***

**Components**

DeviceIdPage.jsx

import React, {useEffect} from "react";

import axios from "axios";

import Cookie from 'js-cookie';

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

import { Card, CardContent, Typography, TextField, makeStyles } from '@material-ui/core';

import MuiThemeProvider from 'material-ui/styles/MuiThemeProvider';

import RaisedButton from 'material-ui/RaisedButton';

import './style/DeviceIdPage.css';

const apiBaseUrl = "http://localhost:9000";

const useStyles = makeStyles(theme => ({

    textField: {

        width: '100%',

        boxSizing: 'border-box',

    },

    root: {

        display: 'block',

        flexGrow: 1

    },

    menuButton: {

        marginRight: theme.spacing(2),

    },

    title: {

        flexGrow: 1,

    }

  }));

export const DeviceIdPage = () => {

    const classes = useStyles();

    const [text1,setText1] = React.useState("2015-01-01 23:00:00");

    const [text2,setText2] = React.useState("2020-01-02 23:00:00");

    const [users, setUsers] = React.useState([]);

    const [showPositions, shouldShowPositions] = React.useState(false);

    useEffect(() => {

      axios.get(apiBaseUrl)

    .then((response)=> {

        setUsers(response.data);

    })

    .catch(() => {

        console.log('Unknown error');

    })}, []);

    const getDates = () => users.filter((user)=> text1 < user.date && user.date < text2)

    return (

        <div>

        <MuiThemeProvider key={"theme"}>

            <TextField className={classes.textField} label="Start Date" style={style}

defaultValue={text1} onChange={(event)=> {

                shouldShowPositions(false);

                setText1(event.target.value)}}>{text1}</TextField>

            <TextField className={classes.textField} label="End Date" style={style}

defaultValue={text2} onChange={(event)=> {

                shouldShowPositions(false);

                setText2(event.target.value)}}>{text2}/></TextField>

            <RaisedButton label="Get Positions" primary={true} style={style} onClick={()=> shouldShowPositions(true)}  />

            <Route render={({history}) =>

            <span>

            <RaisedButton label="Show on Map" primary={true} style={style} onClick={() => history.push("/map")} />

            <div class="button">

            <RaisedButton  label="Logout" secondary={true} onClick={()=> {

                Cookie.remove("token");

                history.push("/")

            }} />

            </div>

            </span>

            }/>

        </MuiThemeProvider>

        {showPositions && getDates().map((user)=>

        <Card>

            <CardContent>

            <Typography variant="h6" component="div">

                    Lat:

                <Typography variant="body" component="span">

                    <b>{user.lat.toFixed(2)}</b>

                </Typography>

            </Typography>

            <Typography variant="h6" component="div">

                    Long:

                    <Typography variant="body" component="span">

                    <b>{user.lng.toFixed(2)}</b>

                </Typography>

            </Typography>

            <Typography variant="h6" component="div">

                    Data:

                    <Typography variant="body" component="span">

                    <b>{user.date}</b>

                </Typography>

            </Typography>

            </CardContent>

        </Card>)

        }

        {showPositions && getDates().length > 0 && <RaisedButton label="Clear" secondary=

{true} style={style} onClick={()=> shouldShowPositions(false)}/>}

        </div>

    );

}

const style = {

    margin: 30,

};

export default DeviceIdPage;

LoginForm.jsx

import React, {useState} from 'react';

import MuiThemeProvider from 'material-ui/styles/MuiThemeProvider';

import AppBar from 'material-ui/AppBar';

import Link from '@material-ui/core/Link';

import RaisedButton from 'material-ui/RaisedButton';

import TextField from 'material-ui/TextField';

import Cookie from 'js-cookie';

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

import axios from 'axios';

const apiBaseUrl = "http://localhost:8000";

const inOneHour = 1/24;

const LoginForm = () => {

  const [email,setEmail] = useState("");

  const [password,setPassword] = useState("");

  const [isLoggedIn, setLoggedIn] = useState(Cookie.get("token") !== undefined);

  const handleClick = () => {

    axios.get(apiBaseUrl)

   .then((response)=> {

    if(response.data.filter((user)=> user.email === email)[0])

      if(response.data.filter((user)=> user.password === password)[0]){

        Cookie.set("token", response.data.filter((user)=> user.password === password)[0].id, { expires: inOneHour });

        setLoggedIn(true);

      }

      else alert("Wrong password!");

    else alert("Username doesn't exist!");

   })

   .catch(() => {

     console.log('Unknown error');

   });

  }

  return (

      <div>

        <MuiThemeProvider>

        <AppBar

             title="GPS APP"

         />

        </MuiThemeProvider>

        <MuiThemeProvider>

        <div>

        </div>

        </MuiThemeProvider>

        <MuiThemeProvider key={"theme"}>

      <div>

      <TextField

        hintText="Enter your email"

        floatingLabelText="Email"

        required

        onChange = {(event) => setEmail(event.target.value)}

        />

      <br/>

        <TextField

        type="password"

        hintText="Enter your Password"

        floatingLabelText="Password"

        required

        onChange={(event) => setPassword(event.target.value)}

          />

        <br/>

        <RaisedButton label="Login" primary={true} style={style} onClick={(event) => handleClick(event)}/>

        <Route render={({history}) => {

        isLoggedIn && history.push("/device")

        return (

          <Link href="" onClick={() => history.push("/register")}>

            Don't have an account?Sign up.

          </Link>

        )}}/>

     </div>

      </MuiThemeProvider>

    </div>

        );

}

const style = {

  margin: 15,

};

export default LoginForm;

MapComponent.jsx

import React, {useEffect} from "react";

import axios from "axios";

import Cookie from "js-cookie";

import MapContainer from '../containers/MapContainer';

const apiBaseUrl = "http://localhost:9000";

const Map = () => {

  const [disabled, setDisabled] = React.useState(false);

  const [locations, setLocations] = React.useState(undefined);

    useEffect(() => {

      setDisabled(false);

      axios.get(apiBaseUrl)

    .then((response)=> {

      !!response.data.filter(location => location.id === Cookie.get("token")).length && setDisabled(true)

      setLocations(response.data);

    })

    .catch(() => {

        console.log('Unknown error');

    })}, []);

    return (

      !!locations ?

      <div>

        <MapContainer

          googleMapURL="https://maps.googleapis.com/maps/api/js?key=AIzaSyCKT9LmoztX4FjDThD6puBbkIfAstQYZdk"

          loadingElement={<div style={{ height: `100%` }} />}

          containerElement={<div style={{ height: `400px` }} />}

          mapElement={<div style={{ height: `100%` }} />}

          center={{lat: 46.23, lng: 27.54}}

          zoom={8}

          disabledButton={disabled}

          locations={locations}

        />

      </div>

      :

      <div>Could not get data</div>

    );

  };

  export default Map;

RegisterForm.jsx

  import React, { useState} from 'react';

  import AppBar from 'material-ui/AppBar';

  import RaisedButton from 'material-ui/RaisedButton';

  import Button from '@material-ui/core/Button';

  import TextField from 'material-ui/TextField';

  import axios from 'axios';

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

  const RegisterForm = () => {

    const [username,setUsername] = useState("");

    const [password,setPassword] = useState("");

    const [passwordConfirmation,setPasswordConfirmation] = useState("");

    const [email,setEmail] = useState("");

    const handleSubmit = () => {

            const apiBaseUrl = "http://localhost:8000/";

            const regex = /^(([^<>()\[\]\\.,;:\s@"]+(\.[^<>()\[\]\\.,;:\s@"]+)\*)|(".+"))@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}])|(([a-zA-Z\-0-9]+\.)+[a-zA-Z]{2,}))$/;

            if(email.length > 5){

                if(username.length > 5) {

                    if(password.length > 5)

                    {

                        if(regex.test(email)){

                          if(password === passwordConfirmation)

                          {

                            axios.post(apiBaseUrl+ "insert?username="+username+"&email="+email+"&password="+password)

                            .then(function (response) {

                            if(response.data.code === 200){

                              console.log("Succes");

                            }

                            })

                            .catch(() =>  {

                            console.log('Unknown error');

                            });

                          }

                          else

                            alert("Password mismatch");

                        }

                        else alert('Invalid email');

                      }

                    else{

                      alert("Password too short");

                    }

                  }

                  else{

                    alert("Email too short");

                  }

                }

                else{

                  alert("Username too short");

                }

             }

        return (

            <div>

             <AppBar

              title="Register"

             >

               <Route render={({history}) => <Button style={{color:"white"}} onClick={() => history.push("/")}>

                 Login

              </Button>}/>

              </AppBar>

                <form

                  onSubmit={()=> handleSubmit()}

                  autoComplete="off"

                >

                    <br/>

                    <TextField

                    hintText="Enter your username"

                    floatingLabelText="Username"

                    onChange = {(event) => setUsername(event.target.value)}

                    required

                    />

                    <br/>

                    <TextField

                    hintText="Enter your email"

                    floatingLabelText="Email"

                    onChange = {(event) => setEmail(event.target.value)}

                    required

                    />

                    <br/>

                    <TextField

                    type = "password"

                    hintText="Enter your password"

                    floatingLabelText="Password"

                    onChange = {(event) => setPassword(event.target.value)}

                    required

                    />

                    <br/>

                    <TextField

                    type = "password"

                    hintText="Confirm passowrd"

                    floatingLabelText="Password"

                    onChange = {(event) => setPasswordConfirmation(event.target.value)}

                    required

                    />

                    <br/>

                    <RaisedButton label="Register" primary={true} style={style} type="submit" variant="outlined"/>

                    <br/>

                </form>

            </div>

            );

    }

  const style = {

    margin: 15,

  };

  export default RegisterForm;

style/DeviceIdPage.css

.button {

    position: absolute;

    bottom: 0px;

    right: 0px;

}

**Containers**

MapContainer.jsx

import React from "react";

import axios from "axios";

import Cookie from "js-cookie";

import Button from '@material-ui/core/Button';

import RaisedButton from 'material-ui/RaisedButton';

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

import { getCurrentDate } from "../utils/DateGetter";

import {

  withGoogleMap,

  GoogleMap,

  Marker,

  withScriptjs

} from "react-google-maps";

import './style/MapContainer.css'

const apiBaseUrl = "http://localhost:9000/";

const Markers = ({ locations }) => {

  return locations.map(pos => {

    return (

      <Marker position={{ lat: pos.lat, lng: pos.lng }} />

    );

  });

};

const Map = ({ locations, zoom, center }) => {

  return (

    <GoogleMap defaultZoom={zoom} defaultCenter={center}>

      <Markers locations={locations} />

    </GoogleMap>

  );

};

const getRandomInRange = (from, to, fixed) => {

  return (Math.random() \* (to - from) + from).toFixed(fixed) \* 1;

};

class MapWithMarker extends React.Component {

  constructor(props) {

    super(props);

    this.state = { disabledButton: props.disabledButton, locations: props.locations };

  }

  // componentWillUnmount(){

  //   Cookie.remove("token");

  // }

  addMarker = () => {

    const newLocation = {

      id: Cookie.get("token"),

      date: getCurrentDate(),

      lat: getRandomInRange(43, 48, 2),

      lng: getRandomInRange(20, 29, 2)

    };

    this.setState(prevState => ({

      disabledButton: true, locations: [...prevState.locations, newLocation]

    }));

      axios.post(apiBaseUrl+ "insert?id="+newLocation.id+"&date="+newLocation.date+"&lat="+newLocation.lat+"&lng="+newLocation.lng)

          .then(function (response) {

          if(response.data.code === 200){

              console.log("Succes");

          }})

          .catch(() =>  {

              console.log('Unknown error');

      });

  }

  removeMyMarker = () => {

    const currentId = Cookie.get("token");

      this.state.locations.forEach((user)=>

      user.id === currentId &&

      axios.post(apiBaseUrl+ "delete?id="+user.id)

          .then(() => {})

          .catch(() =>  {})

      );

    this.setState({disabledButton: false, locations: this.state.locations.filter(user =>

user.id !== currentId)})

  }

  render() {

    return (

      <div>

            <div>

              <Map

                center={this.props.center}

                zoom={this.props.zoom}

                locations={this.state.locations}

              />

              <Button variant="outlined" disabled={this.state.disabledButton}

color="primary" onClick={()=> this.addMarker()}>Add static marker</Button>

              <Button variant="outlined" disabled={!this.state.disabledButton}

color="secondary" onClick={()=> this.removeMyMarker()}>Remove my maker</Button>

              <Route render={({history}) => <div class="button"><RaisedButton

label="Go Back" primary={true} onClick={() =>{

                history.push("/device")}

                } />

               <RaisedButton label="Logout" secondary={true} onClick={()=> {

                Cookie.remove("token");

                history.push("/")

                }} />

            </div>}/>

            </div>

      </div>

    );

  }

}

export default withScriptjs(withGoogleMap(MapWithMarker));

style/MapContainer.css

.button {

    position: fixed;

    margin: 10px;

    bottom: 0px;

    right: 0px;

}

**Handlers**

AppHandler.jsx

import React from "react";

import {

  BrowserRouter as Router,

  Switch,

  Route,

} from "react-router-dom";

import {PrivateRoute} from "../utils/PrivateRoute";

import LoginForm from '../components/LoginForm';

import RegisterForm from '../components/RegisterForm';

import DeviceIdPage from "../components/DeviceIdPage";

import MapComponent from '../components/MapComponent';

import MuiThemeProvider from 'material-ui/styles/MuiThemeProvider';

export const RoutesHandler = () =>{

    return(

   <Router>

      <div>

        <MuiThemeProvider key={"theme"}>

          <Switch>

            <Route path="/register" component={RegisterForm}>

             <RegisterForm/>

            </Route>

            <PrivateRoute path="/device" component={DeviceIdPage}>

              <DeviceIdPage/>

            </PrivateRoute>

            <PrivateRoute path="/map" component={MapComponent}>

              <MapComponent/>

            </PrivateRoute>

            <Route path="/" component={LoginForm}>

              <LoginForm/>

            </Route>

          </Switch>

        </MuiThemeProvider>

      </div>

    </Router>

    );

}

export default RoutesHandler;

**Utils**

DateGetter.jsx

export const getCurrentDate = () => {

    const today = new Date();

    const year = today.getFullYear();

    const month = (today.getMonth()+1) < 10 ? '0'+(today.getMonth()+1) : (today.getMonth()+1);

    const day = today.getDate() < 10 ? '0' + today.getDate() : today.getDate();

    const hours = today.getHours() < 10 ? '0' + today.getHours() : today.getHours();

    const minutes = today.getMinutes() < 10 ? '0' + today.getMinutes() : today.getMinutes();

    const seconds = today.getSeconds() < 10 ? '0' + today.getSeconds() : today.getSeconds();

    const date = year+'-'+ month +'-'+ day;

    const time = hours + ":" + minutes + ":" + seconds;

    const dateTime = date+' '+time;

    return dateTime;

}

IdGenerator.jsx

export const idGenerator = () => '\_' + Math.random().toString(36).substr(2, 9);

PrivateRoute.jsx

import React from 'react';

import Cookie from 'js-cookie';

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

export const PrivateRoute = ({ component: Component, ...rest }) =>

    Cookie.get("token") ?

    <Route {...rest} render={(props) => {

        return(

            Cookie.get("token")

                ? <Component {...props} />

                : <Redirect to='/' />

            )}

        } />

        :

    <Redirect to='/'/>

**APP**

App.jsx

import React from "react";

import AppHandler from "./handlers/AppHandler";

export const App = () => {

    return (

        <div>

       <AppHandler/>

       </div>

    )

}

export default App;