Michał Tyka – 299598
Internet Engineering – IS WIMiIP

The following project is a SPA (Single Page Application) based on an already provided REST API service, created using ReactJS. My application provides a convenient way to view schedule timetables with detailed specifics about particular conference such as: title, starting time, place and authors. It also allows you to download files and papers relevant to the meeting. Clicking on any day of the week displays only conferences happening on that particular day of current week.

Users can create an account, log-in and also view account information.

1.Routing

A *react-router-dom* library is being used for routing. It provides components which are needed for routing.

```
15      <BrowserRouter>
16        <Switch>
17          <Route exact path={'/'} component={Start}></Route>
18          <Route path='/register' component={Register}></Route>
19          <Route path='/login' component={Login}></Route>
20          <Route path='/unauthorized' component={ErrorC}></Route>
21          <Route path='/account' component={Account}></Route>
22          <Route path='' component={NotFound}></Route>
23        </Switch>
24      </BrowserRouter>
```

Available routes:
1. */register* - register page
2. */login* - login page
3. */unauthorized* - user is being redirected to this page when trying to access unauthorized data.
4. /account - account page
5. */* - (index page) - main page which displays conference schedule
When the route is not found, the user is being redirected to the NotFound route.

2.API Communication

Axios - promise based HTTP client for browser and node.js is used as fetch API in my application.

First an instance of axios, with base URL for external API, is created and exported for further use.

```
1    import axios from 'axios';
2
3    export default axios.create({
4      baseURL: `https://ie2020.kisim.eu.org/api`,
5    });
```

Then when it is needed, this instance is imported and appropriate method is called (e.g. .post(), .get()) with additional parameters if necessary.

```
24        api
25          .get(`/users/me`, { headers: { Authorization: `Bearer ${token}` } })
26          .then((res) => {
27            console.log(res.data);
28            setData({
29              id: res.data.id,
30              email: res.data.email,
31              createdAt: res.data.createdAt,
32            });
33          })
34          .catch((err) => {
35            console.log(err);
36          });
```

3.JSON Web token storage

JSON web token is received upon signing in (/auth route) and then saved in the cookie. When necessary, JWT is retrieved from there as shown below.

```
17        const token = Cookies.get('token');
```

```
38            Cookies.set('token', res.data.token);
```

4.Validation

Validation is based on HTML input attributes such as: "required" and "type". Server API provides further validation as it returns a specific error message upon sending invalid data.


```
68   <Form.Control
69       required
70       type='email'
71       placeholder='Enter email'
72       onChange={(e) => {
73         setEmail(e.target.value);
74       }}
75   />
```
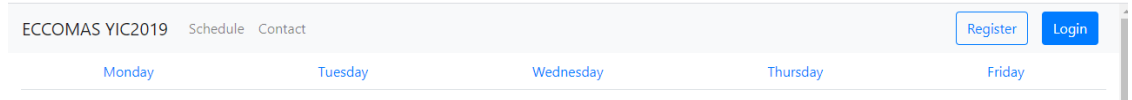
5.Github repository

https://github.com/KISiM-AGH/Micha--Tyka-299598-ie-2020

6.Components

Application is composed of following main components:

a. Layout

Component containing navigation. Uses React-bootstrap Navbar component.

b. AccountSettings

Component displaying current user's data. Data is fetched using /users/me API endpoint.

If user is logged in:
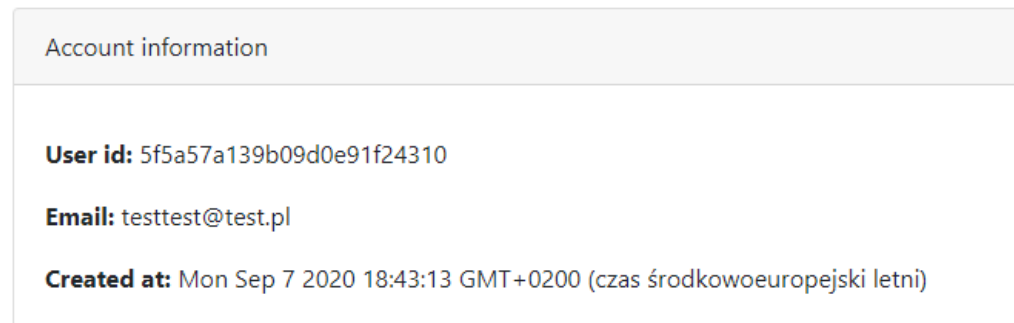
Otherwise it redirect user to /unauthorized and displays component ErrorMessage

c. ErrorMessage

Component visible on /unauthorized route informing user that authorisation is required. Uses React-bootstrap Jumbotron component.

Unauthorized

Request has not been applied because it lacks valid authentication credentials for the target resource.

Go back
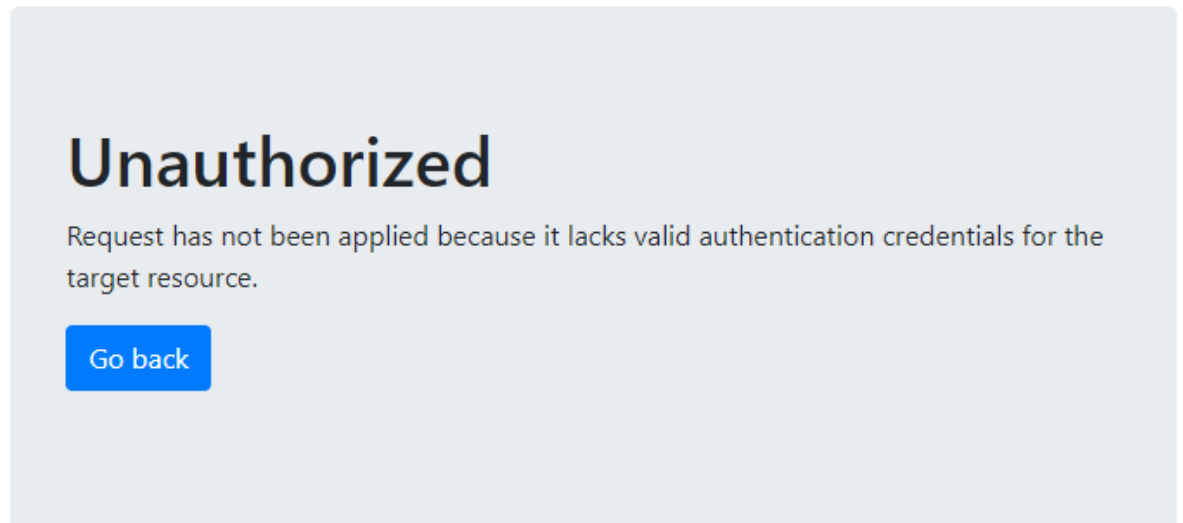
d. RegisterForm

Component visible on /register route. Uses React-bootstrap Card and Form group components. Accepting terms of service (TOS) is mandatory and required by html using required attribute. Component displays errors received from the server.

Register

Email address

Enter email

Password

Password

☐ Accept TOS

Submit

email must match regular expression /^\S+@\S+\.\S+$/

password must have length greater than or equal to 6

e. LoginForm

Component visible on /login route. Uses React-bootstrap Card and From group components.

f. NotFoundMessage

Component visible when route is not found. Uses React-bootstrap Jumbotron component.

g. ScheduleDayList (ScheduleCard)
Component used to display singular conference. Uses React-bootstrap Card component.

h. ScheduleList

Component used to display list of conferences. Uses React-bootstrap CardColumns component.

Monolithic Solution of Mixed-dimensional Models in Fractured Porous Media by Multigrid Methods	The Concept of a Virtual Ring for Mesh Update Methods in Moving Boundary Problems	Unfitted space-time finite element methods for problems involving moving domains	VMS Implementation for CBRN Dispersion Modeling in the Atmosphere
Starts at: 07:30	Starts at: 13:20	Starts at: 08:40	Starts at: 09:20
Authors: Carmen Rodrigo	**Authors:** Fabian Key / Daniel Hilger / Stefanie Elgeti	**Authors:** Christoph Lehrenfeld	**Authors:** Yoshiyuki Nishio / Bart Janssens / Karim Limam / Jeroen Van Beeck
Place: Building B8 / Rooms A + B	**Place:** Building B4 / Room C	**Place:** Building B8 / Room B	**Place:** Building B8 / Room A
Session: Plenary Session	**Session:** Boundary-Conforming Discretization Methods for Complex Moving Domains	**Session:** Recent Advances in Space-Time Methods	**Session:** Novel Numerical Methods for Fluid-Structure Interaction Problems
Keywords: *Multigrid Fractured Porous Media Discrete Fracture Models Mixed-dimensional Models*	**Keywords:** *CFD Space-Time FEM Moving Boundaries Mesh Update Method Virtual Ring Packaging Machines*	**Keywords:** *Space-Time FEM Numerical Integration Finite element method Isoparametric FEM Level Set methods Unfitted FEM CutFEM*	**Keywords:** *ABL CFD FEM LES Multiphase Flow Turbulent Flow*
Download Paper	Download Paper	Download Paper	Download Paper

7.State handling

Application uses function components and most components use state handling provided by *useState()* hook. Also if necessary some data is being passed as props to another components.

```
8    const Layout = (props) => {
9      const [isLogged, setIsLogged] = useState(false);
10     const [username, setUsername] = useState('user');
11     const [needsRefresh, setNeedsRefresh] = useState(false);
```

8.Graphical library

React-Bootstrap in version 1.3.0 was used to create both app wireframe and provide styles for each component. React-bootstrap is a library adapting bootstrap 4.5 - the most popular html/css/js library to work in react environment. It doesn't require slow and outdated jQuery as it was written from the ground up.

To further improve visual experience *styled-icons* icon library was used to provide icons for amongst others download icon for conference paper:
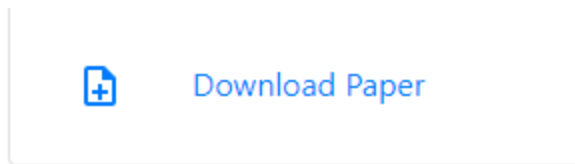
Library *react-loading-skeleton* was used in order to provide loading animations while asynchronous request is preformed and response is being awaited or to handle situations where response fails with header x-ratelimit-remaining: 0

9.Worth mentioning

Due to using React-bootstrap application is responsive and mobile friendly.
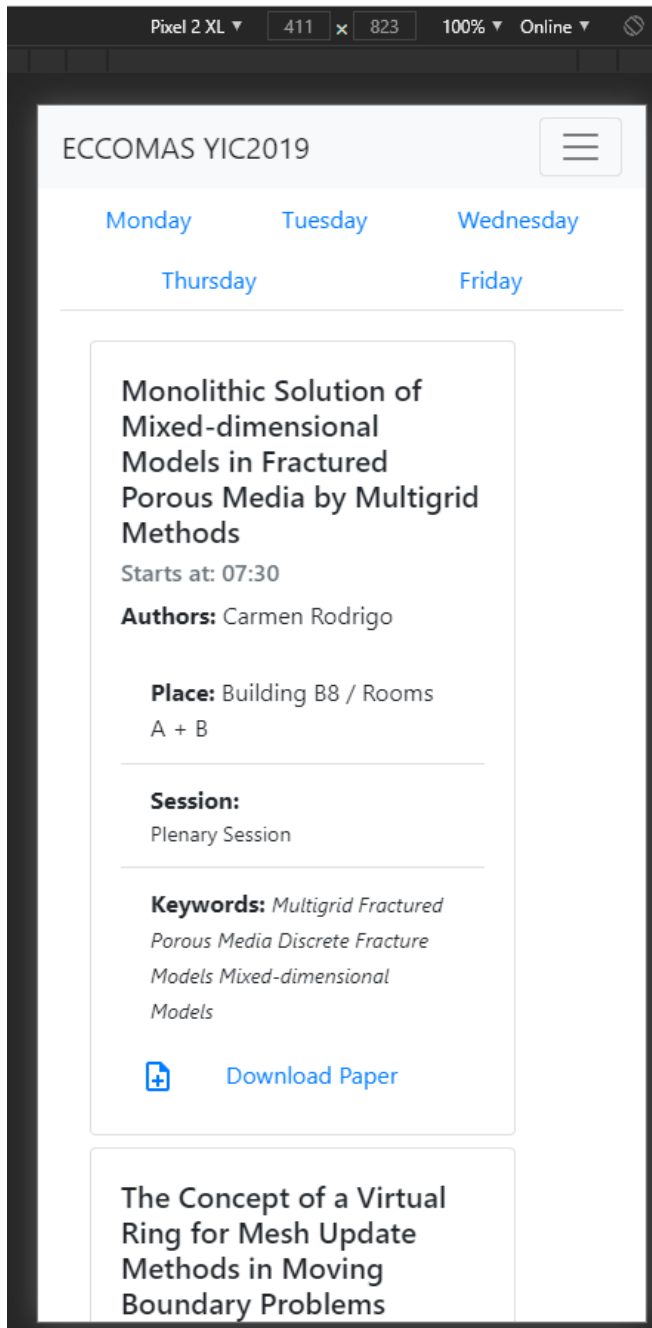
Used dependencies:

React, Axios, react-bootstrap, js-cookie