DEPARTMENT OF CSE
COURSE CODE: 23SDCS12A / 23SDCS12R
FULL STACK APPLICATION DEVELOPMENT

Date of the Session: __/__/_____ **Time of The Session:_____to_____**

LAB - 10 ➔ Implementing Authentication and Role Based Access

Prerequisites:

Implementation skill in Reactjs

Implementation skill in Spring Boot Application with JPA and Database

Exercise:

Implement the necessary ReactJS front end pages (signin, signup, home, admin) for sending the request and receiving the responses from the back end (spring boot app) designed to handle signin and signup request with database integration.

During registering the user, get input as admin or user. This is to store the user role in database. Have logout button in the front end to sign-out the authentication.

While login, permit the user or admin to Home page on successful login. Permit the admin to admin page and not the user.

❖ Watch The Video And Do In Eclipse Workspace

10 https://youtu.be/4RdQChNNowo?si=Pzyt4AvtJ_x21DsT

Frontend

Signin.jsx

```
import React from "react";
import './style.css';
import axios from 'axios';

export default function Signin({store}) {

function checkUser() {

   var un = document.getElementsByName("un")[0].value;
   var pw = document.getElementsByName("pw")[0].value;

axios.post("http://localhost:8081/check", {

   "un": un,
   "pw": pw,
}).then((res)=>{
   alert(res.data)
   if(res.data !== "0"){
sessionStorage.setItem("un" ,un)
sessionStorage.setItem("role" ,res.data)
store.dispatch({ "type": "page", "data": "Home" })
   }
})
}
   return (
      <div>
         <center>
            <div className='signin-box'>
               <table>
                  <tbody>
                     <tr>
```

```
                    <td colSpan={2}>Signin Page</td>
                  </tr>
                  <tr>
                    <td>Username:</td>
                    <td><input type="text" name="un" className="form-control" /></td>
                  </tr>
                  <tr>
                    <td>Password:</td>
                    <td><input type="password" name="pw" className="form-control"
/></td>
                  </tr>
                  <tr>
                    <td colSpan={2}><button onClick={checkUser}>Login</button></td>
                  </tr>
                </tbody>
              </table>
            </div>
          </center>
        </div>
    );
}
```

Signup.jsx

```
import React from "react";
import './style.css';
import axios from 'axios';

export default function Signup() {

function insertUser(){

   var un = document.getElementsByName("un")[0].value;
   var pw = document.getElementsByName("pw")[0].value;
   var role = document.getElementsByName("role")[0].value;

axios.post("http://localhost:8081/user", {

   "un": un,
```

```
    "pw": pw,
    "role": role
  }).then((res)=>{
    alert(res.data)

  })
  }

    return (
      <div>
        <center>
        <div className='signin-box'>
          <table>
            <tbody>
              <tr>
                <td colSpan={2}>Signup Page</td>
              </tr>
              <tr>
                <td>Username:</td>
                <td><input type="text" name="un"class="form-control" /></td>
              </tr>
              <tr>
                <td>Password:</td>
                <td><input type="password" name="pw" class="form-control"  /></td>
              </tr>
              <tr>
                <td>Role:</td>
                <td>
                  <select class="form-select" name="role">
                    <option value={1}>Admin</option>
                    <option value={2}>User</option>
                  </select>
                </td>
              </tr>
              <tr>
                <td colSpan={2}><button onClick={insertUser}> Signup </button></td>
              </tr>
            </tbody>
          </table>
```

```
        </div>
        </center>
      </div>
   );
}
```

Show.jsx

```
import React from "react";
import Signin from "./Signin";
import Signup from "./Signup";
import Home from "./Home";
import Admin from "./Admin";
import Error from "./Error";
import ResponsiveAppBar from "./Appbar";
import './style.css';

export default function Show({store}) {

   function Display(){

      switch(store.getState()){
         case "Signin":
            return (<div> <Signin store = {store}/> </div>)
            case "Signup":
            return(<div> <Signup /> </div>)

            case "Home":
               if(sessionStorage.getItem("role") === "1" || sessionStorage.getItem("role")
=== "2")
            return(<div> <Home /> </div>)
               else
            return(<div> <Error /> </div>)

            case "Admin":
               if(sessionStorage.getItem("role") === "1")
            return(<div> <Admin /> </div>)
               else
            return(<div> <Error /> </div>)
```

```
      }
    }
    return (
      <div className='parent'>
        <div className='menu-bar'>
        <ResponsiveAppBar store = {store} />

        </div>
        <div className='page'>
          <Display />
        </div>
      </div>
    );
}
```

Home.jsx

```
import React from "react";

export default function Home() {
  return (
    <div>
      <center>
        <div className='signin-box'>
          This is a Home Page
        </div>
      </center>
    </div>
  );
}
```

Error.jsx

```
import React from "react";

export default function Error() {
   return (
      <div>
         <center>
            <div className='signin-box'>
               This is an Error Page. You are not authorized to access other pages.
            </div>
         </center>
      </div>
   );
}
```

Appbar.jsx

```
import * as React from 'react';
import AppBar from '@mui/material/AppBar';
import Box from '@mui/material/Box';
import Toolbar from '@mui/material/Toolbar';
import IconButton from '@mui/material/IconButton';
import Typography from '@mui/material/Typography';
import Menu from '@mui/material/Menu';
import MenuIcon from '@mui/icons-material/Menu';
import Container from '@mui/material/Container';
import Avatar from '@mui/material/Avatar';
import Button from '@mui/material/Button';
import Tooltip from '@mui/material/Tooltip';
import MenuItem from '@mui/material/MenuItem';
import AdbIcon from '@mui/icons-material/Adb';

const pages = ['Signin', 'Signup', 'Home', 'Admin'];
const settings = ['Logout'];

function ResponsiveAppBar({store}) {
  const [anchorElNav, setAnchorElNav] = React.useState(null);
```

```
  const [anchorElUser, setAnchorElUser] = React.useState(null);

  const handleOpenNavMenu = (event) => {
   setAnchorElNav(event.currentTarget);
  };
  const handleOpenUserMenu = (event) => {
   sessionStorage.setItem("un", "null")
   sessionStorage.setItem("role", "0")
   store.dispatch({ type: "page", data: "Signup" });

   setAnchorElUser(event.currentTarget);
  };

  const handleCloseNavMenu = (event) => {
   store.dispatch ({"type": "page" , "data": event.currentTarget.getAttribute("cp")})
   setAnchorElNav(null);
  };

  const handleCloseUserMenu = () => {
   sessionStorage.setItem("un", "null")
   sessionStorage.setItem("role", "0")
   store.dispatch({ type: "page", data: "Signup" });

   setAnchorElUser(null);
  };

  return (
   <AppBar position="static">
    <Container maxWidth="xl">
     <Toolbar disableGutters>
      <AdbIcon sx={{ display: { xs: 'none', md: 'flex' }, mr: 1 }} />
      <Typography
       variant="h6"
       noWrap
       component="a"
       href="#app-bar-with-responsive-menu"
       sx={{
        mr: 2,
        display: { xs: 'none', md: 'flex' },
```

```
    fontFamily: 'monospace',
    fontWeight: 700,
    letterSpacing: '.3rem',
    color: 'inherit',
    textDecoration: 'none',
   }}
  >
   LOGO
  </Typography>

  <Box sx={{ flexGrow: 1, display: { xs: 'flex', md: 'none' } }}>
   <IconButton
    size="large"
    aria-label="account of current user"
    aria-controls="menu-appbar"
    aria-haspopup="true"
    onClick={handleOpenNavMenu}
    color="inherit"
   >
    <MenuIcon />
   </IconButton>
   <Menu
    id="menu-appbar"
    anchorEl={anchorElNav}
    anchorOrigin={{
     vertical: 'bottom',
     horizontal: 'left',
    }}
    keepMounted
    transformOrigin={{
     vertical: 'top',
     horizontal: 'left',
    }}
    open={Boolean(anchorElNav)}
    onClose={handleCloseNavMenu}
    sx={{ display: { xs: 'block', md: 'none' } }}
   >
    {pages.map((page) => (
     <MenuItem key={page} cp = {page} onClick={handleCloseNavMenu}>
```

```
            <Typography sx={{ textAlign: 'center' }}>{page}</Typography>
          </MenuItem>
        ))}
      </Menu>
    </Box>
    <AdbIcon sx={{ display: { xs: 'flex', md: 'none' }, mr: 1 }} />
    <Typography
      variant="h5"
      noWrap
      component="a"
      href="#app-bar-with-responsive-menu"
      sx={{
        mr: 2,
        display: { xs: 'flex', md: 'none' },
        flexGrow: 1,
        fontFamily: 'monospace',
        fontWeight: 700,
        letterSpacing: '.3rem',
        color: 'inherit',
        textDecoration: 'none',
      }}
    >
      LOGO
    </Typography>
    <Box sx={{ flexGrow: 1, display: { xs: 'none', md: 'flex' } }}>
      {pages.map((page) => (
        <Button
          key={page}
          cp = {page}
          onClick={handleCloseNavMenu}
          sx={{ my: 2, color: 'white', display: 'block' }}
        >
          {page}
        </Button>
      ))}
    </Box>
    <Box sx={{ flexGrow: 0 }}>
      <Tooltip title="Open settings">
        <IconButton onClick={handleOpenUserMenu} sx={{ p: 0 }}>
```

```
            <Avatar alt= {sessionStorage.getItem("un")} src="/static/images/avatar/2.jpg"
/>
          </IconButton>
        </Tooltip>
        <Menu
          sx={{ mt: '45px' }}
          id="menu-appbar"
          anchorEl={anchorElUser}
          anchorOrigin={{
            vertical: 'top',
            horizontal: 'right',
          }}
          keepMounted
          transformOrigin={{
            vertical: 'top',
            horizontal: 'right',
          }}
          open={Boolean(anchorElUser)}
          onClose={handleCloseUserMenu}
        >
          {settings.map((setting) => (
            <MenuItem key={setting} onClick={handleCloseUserMenu}>
              <Typography sx={{ textAlign: 'center' }}>{setting}</Typography>
            </MenuItem>
          ))}
        </Menu>
      </Box>
    </Toolbar>
  </Container>
 </AppBar>
 );
}
export default ResponsiveAppBar;
```

Admin.jsx

```
import React from "react";

export default function Admin() {
   return (
      <div>
         <center>
            <div className='signin-box'>
               This is an Admin Page
            </div>
         </center>
      </div>
   );
}
```

Navreducer.jsx

```
import React from "react";

export default function Navreducer (state="Signup", action){

   switch(action.type)
   {
      case "page":
         state = action.data
         return state
         default:
            return state

   }
}

//action = {"type": "page" , "data": "Home"}
```

Backend

UserRepo.java

```
package com.klu;

import org.springframework.data.jpa.repository.JpaRepository;

public interface UserRepo extends JpaRepository<User, String> {

	public User findByUn (String un);
}
```

User.java

```
package com.klu;

import jakarta.persistence.Entity;
import jakarta.persistence.Id;
@Entity
public class User {

@Id
String	un;
String	pw;
String	role;
public String getUn() {
	return un;
}
public void setUn(String un) {
	this.un = un;
}
public String getPw() {
	return pw;
}
public void setPw(String pw) {
```

```
        this.pw = pw;
}
public String getRole() {
        return role;
}
public void setRole(String role) {
        this.role = role;
}
@Override
public String toString() {
        return "User [un=" + un + ", pw=" + pw + ", role=" + role + ", getUn()=" + getUn() + ",
getPw()=" + getPw()
                        + ", getRole()=" + getRole() + "]";
}

}
```

Service.java

```
package com.klu;

import org.springframework.beans.factory.annotation.Autowired;
import java.util.List;

@org.springframework.stereotype.Service
public class Service {

   @Autowired
   private ProductRepo repo1;

   @Autowired
   UserRepo repo2;

   public String insertUser (User user) {

        repo2.save(user);
        return "Success";
   }
```

```
public String checkUser(User user) {

    User userRetrieved = repo2.findByUn(user.getUn());
if( userRetrieved != null) {
    if(userRetrieved.getPw().equals(user.getPw())) {
return userRetrieved.getRole();
    }

    else {
            return "0";
    }
}

else {
    return "0";
}
}

public String insertData(Product product) {
   repo1.save(product);
   return "Inserted Successfully";
}

public String updateData(Product product) {
   if (repo1.findById(product.getId()) != null)
    repo1.delete(product);
      repo1.save(product);
      return "Updated Successfully";
   }

public String deleteData(int id) {
   repo1.delete (repo1.findById(id) .get());
      repo1.deleteById(id);
      return "Deleted Successfully";
}

public List<Product> retrieveData() {
   return repo1.findAll();
}
```

```
}
```

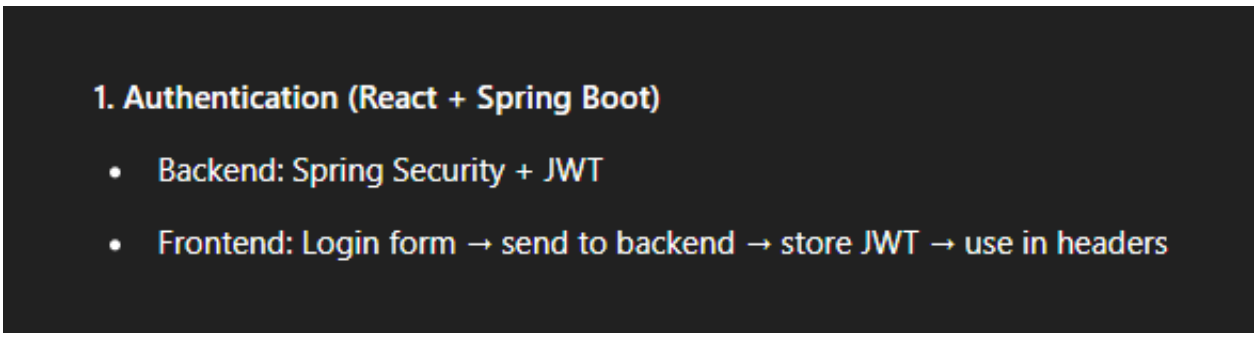
Appcontroller.java

```
package com.klu;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.web.bind.annotation.*;

import java.util.List;

@RestController
@CrossOrigin
public class Appcontroller {

    @Autowired
    Service s;

    // http://localhost:8081/product
    @PostMapping("/product")
    public String insertProduct(@RequestBody Product product) {
        return s.insertData(product);
    }

    // http://localhost:8081/product
    @PutMapping("/product")
    public String updateProduct(@RequestBody Product product) {
        return s.updateData(product);
    }

    // http://localhost:8081/product/1
    @DeleteMapping("/product/{id}")
    public String deleteProduct(@PathVariable int id) {
        return s.deleteData(id);
    }

    @GetMapping("/product")
    public List<Product> retrieveProduct() {
        return s.retrieveData();
```

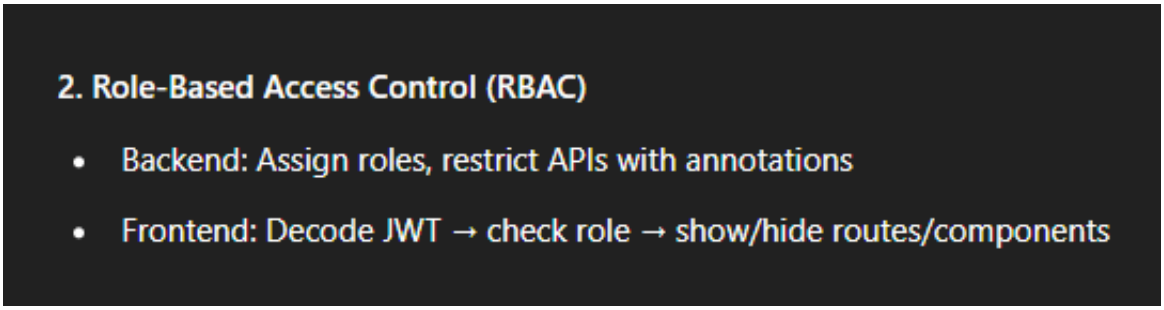
```
    }

    // http://localhost:8081/user
    @PostMapping("/user")
    public String insertUser(@RequestBody User user) {
        return s.insertUser(user);
    }

    // http://localhost:8081/check
    @PostMapping("/check")
    public String retrieveUser(@RequestBody User user) {
        return s.checkUser(user);
    }
}
```

VIVA QUESTIONS:

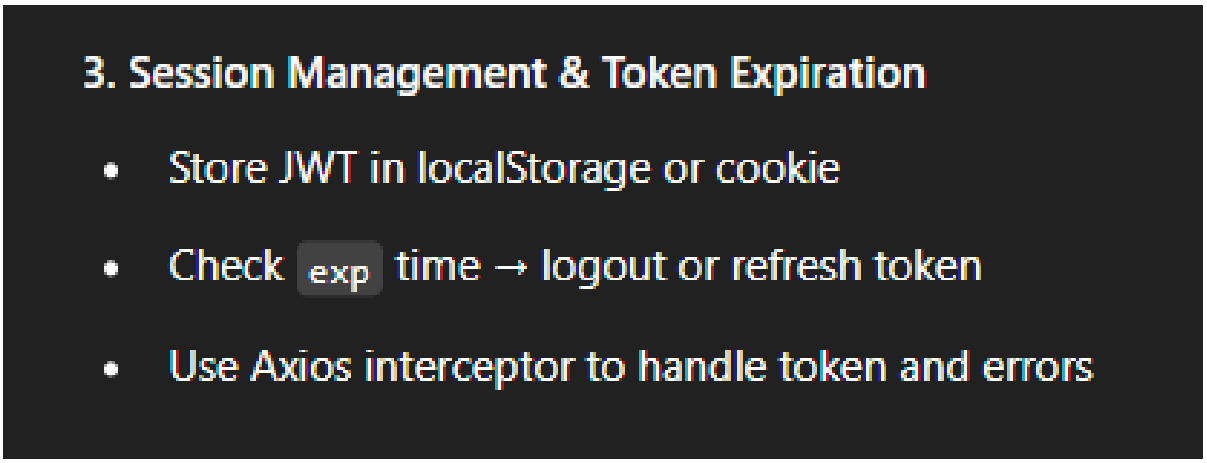
1. How do you implement authentication in a ReactJS and Spring Boot application, and what are the typical steps involved?



2. How do you implement role-based access control (RBAC) in a ReactJS application integrated with a Spring Boot backend?



3. How do you handle session management and token expiration in ReactJS when dealing with authenticated requests?



(For Evaluator's use only)

Comment of the Evaluator (if Any)	Evaluator's Observation Marks Secured _____out of 50 Full Name of the Evaluator: Signature of the Evaluator Date of Evaluation: