**1.AccountController :-**

package com.icinbank.controller;

import java.util.Collections;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.CrossOrigin;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.PutMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RestController;

import com.icinbank.dao.AccountRepository;

import com.icinbank.dao.SaccountRepository;

import com.icinbank.details.TransactionDetails;

import com.icinbank.details.TransferDetails;

import com.icinbank.model.Account;

import com.icinbank.model.Saccount;

import com.icinbank.model.Transfer;

import com.icinbank.model.UserHistory;

import com.icinbank.response.DepositResponse;

import com.icinbank.response.TransferResponse;

import com.icinbank.response.WithdrawResponse;

import com.icinbank.service.AccountService;

import com.icinbank.service.SaccountService;

import com.icinbank.service.TransferHistoryService;

import com.icinbank.service.UserHistoryService;

@RestController

//@CrossOrigin(origins = "http://localhost:4200")

@CrossOrigin(origins = "\*", allowedHeaders = "\*")

public class AccountController {

    @Autowired

    private AccountService service;

    @Autowired

    private SaccountService sservice;

    @Autowired

    private UserHistoryService hservice;

    @Autowired

    private TransferHistoryService tservice;

    @Autowired

    private AccountRepository adao;

    @Autowired

    private SaccountRepository sdao;

    private final String ifsc="ICIN7465";

    public static boolean isprimary(long *account*) {

        String s = Long.toString(account).substring(0, 10);

        String check="3914918201";

        if(s.equals(check)) {

            return true;

        }

        else

        {

            return false;

        }

    }

    @GetMapping("/account/details/{account}")

    public Account getAccountDetails(@PathVariable("account") int *account* ) {

        return service.getAccountDetails(account);

    }

    @PutMapping("/account/profile")

    public Account updateProfile(@RequestBody Account *account*) {

        return service.updateAccount(account);

    }

    @GetMapping("/account/getprimary/{username}")

    public Account getPrimarydetails(@PathVariable("username") String *username*) {

        return service.getAccount(username);

    }

    @GetMapping("/account/getsaving/{username}")

    public Saccount getSavingdetails(@PathVariable("username") String *username*) {

        return sservice.getAccount(username);

    }

    @PostMapping("/account/deposit")

    public DepositResponse deposit(@RequestBody TransactionDetails *details*) {

        //adao.findByUsername(adao.findByAccno(details.getAccount()).getUsername());

        if(isprimary(details.getAccount())) {

            return service.deposit(details.getAccount(), details.getAmount());

        }

        else {

            return sservice.deposit(details.getAccount(), details.getAmount());

        }

    }

    @PostMapping("/account/withdraw")

    public WithdrawResponse withdraw(@RequestBody TransactionDetails *details*) {

        if(isprimary(details.getAccount())) {

        return service.withdraw(details.getAccount(), details.getAmount());

        }

        else {

            return sservice.withdraw(details.getAccount(), details.getAmount());

        }

    }

    @PostMapping("/account/transfer")

    public TransferResponse transfer(@RequestBody TransferDetails *details*) {

        try {

            if(details.getIfsc().equals(ifsc))

            {

                        Account p=adao.findByUsername(details.getUsername());

                        Saccount s=sdao.findByUsername(details.getUsername());

                        if(p.getAccno()==details.getSaccount() || s.getAccno()==details.getSaccount()) {

                        //String len = Integer.toString(details.getSaccount());

                        if(isprimary(details.getSaccount())) {

                        return service.transfer(details.getSaccount(), details.getRaccount(), details.getAmount());

                        }

                        else

                        {

                            return sservice.transfer(details.getSaccount(), details.getRaccount(), details.getAmount());

                        }

                        }

                        else {

                            TransferResponse response=new TransferResponse();

                            response.setSaccount(details.getSaccount());

                            response.setResponseMessage("Dear user You can only transfer funds from the accounts registed with you");

                            response.setTransferStatus(false);

                            return response;

            }

            }

            else {

                TransferResponse response=new TransferResponse();

                        response.setSaccount(details.getSaccount());

                        response.setResponseMessage("IFSC code is incorrect");

                        response.setTransferStatus(false);

                        return response;

            }

        } catch (Exception *e*) {

            TransferResponse response=new TransferResponse();

            response.setSaccount(details.getSaccount());

            response.setResponseMessage("Please provide an IFSC code");

            response.setTransferStatus(false);

            return response;

        }

    }

    @GetMapping("/account/getHistory/{account}")

    public List<UserHistory> getHistory(@PathVariable("account") long *account* )

    {

        List<UserHistory> history=hservice.getHistory(account);

        Collections.reverse(history);

        return history;

    }

    @GetMapping("/account/getTransfers/{account}")

    public List<Transfer> getTransfers(@PathVariable("account") long *account* )

    {

        return tservice.getTransfers(account);

    }

}

**2. ChequeBookController :-**

package com.icinbank.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.CrossOrigin;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RestController;

import com.icinbank.model.ChequebookRequest;

import com.icinbank.response.ChequeResponse;

import com.icinbank.service.ChequebookService;

@RestController

//@CrossOrigin(origins = "http://localhost:4200")

@CrossOrigin(origins = "\*", allowedHeaders = "\*")

public class ChequeBookController {

    @Autowired

    private ChequebookService service;

    @PostMapping("/cheque/request")

    public ChequeResponse createrequest(@RequestBody ChequebookRequest *chequebook*) {

        return service.createrequest(chequebook);

    }

    @GetMapping("/cheque/getbyAccount/{account}")

    public List<ChequebookRequest> getRequests(@PathVariable("account") long *account*) {

        List<ChequebookRequest> list=service.getRequests(account);

        //Collections.reverse(list);

        //return service.getRequests(account);

        return list;

    }

}

**3. LoginController**

package com.icinbank.controller;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.CrossOrigin;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RestController;

import com.icinbank.details.LoginDetails;

import com.icinbank.response.LoginResponse;

import com.icinbank.service.impl.LoginServiceImpl;

@RestController

//@CrossOrigin(origins = "http://localhost:4200")

@CrossOrigin(origins = "\*", allowedHeaders = "\*")

public class LoginController {

    @Autowired

    LoginServiceImpl service;

    @PostMapping("/login")

    public LoginResponse userLogin(@RequestBody LoginDetails *details*) {

        return service.customerLogin(details);

    }

}

**ProfileController :-**

package com.icinbank.controller;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.CrossOrigin;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.PutMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RestController;

import com.icinbank.details.UpdateDetails;

import com.icinbank.model.User;

import com.icinbank.model.UserDisplay;

import com.icinbank.response.UpdateResponse;

import com.icinbank.service.ProfileService;

@RestController

//@CrossOrigin

@CrossOrigin(origins = "\*", allowedHeaders = "\*")

public class ProfileController {

    @Autowired

    private ProfileService pservice;

    @PutMapping("/profile/update")

    public UpdateResponse updateUser(@RequestBody UpdateDetails *user*) {

        // TODO Auto-generated method stub

        return pservice.updateUser(user);

    }

    @GetMapping("/profile/{username}")

    public User getUser(@PathVariable("username") String *username*) {

        // TODO Auto-generated method stub

        return pservice.getUser(username);

    }

    @GetMapping("home/{username}")

    public UserDisplay userDisplay(@PathVariable("username")String *username*) {

        return pservice.userDisplay(username);

    }

}

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**RegistrationController:-**

package com.icinbank.controller;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.CrossOrigin;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RestController;

import com.icinbank.model.User;

import com.icinbank.response.RegisterResponse;

import com.icinbank.service.RegistrationService;

@RestController

//@CrossOrigin(origins = "http://localhost:4200")

@CrossOrigin(origins = "\*", allowedHeaders = "\*")

public class RegistrationController {

    @Autowired

    private RegistrationService service;

    @PostMapping("/register")

    public RegisterResponse createUser(@RequestBody User *user*) {

        return service.createAccount(user);

    }

}

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**LoginComponent:-**

import { Component, OnInit } from '@angular/core';

import { Router, ActivatedRoute } from '@angular/router';

import { FormBuilder, FormGroup, Validators } from '@angular/forms';

import { Login } from '../\_models';

import { LoginService } from '../login.service';

import { HttpErrorResponse } from '@angular/common/http';

import { AuthService } from '../auth.service';

import Swal from 'sweetalert2';

@Component({

  selector: 'app-login',

  templateUrl: './login.component.html',

  styleUrls: ['./login.component.css']

})

export class LoginComponent implements OnInit {

  loginForm: FormGroup;

  loading = false;

  submitted = false;

  returnUrl: string;

  isLoginError : boolean = false;

  constructor(

    private *formBuilder*: FormBuilder,

    private *route*: ActivatedRoute,

    private *router*: Router,

    private *loginService* : LoginService,

    private *authService* : AuthService

  ) { }

  ngOnInit() {

    this.loginForm = this.formBuilder.group({

      username: ['', Validators.required],

      password: ['', Validators.required]

    });

  }

    // for accessing the form fields

    get fval() { return this.loginForm.controls; }

  onFormSubmit() {

    this.submitted = true;

    if (this.loginForm.invalid) {

      return;

    }

    this.loading = true;

    const result:Login= Object.assign({}, this.loginForm.value);

    this.loginService.loginUser(result.username,result.password).subscribe(

      (*data* : any) =>{

        localStorage.setItem('login',data.loginStatus);

        localStorage.setItem('username',data.username);

        if(data.loginStatus==true){

          Swal.fire({

            icon: 'success',

            title: 'Login successful',

            showConfirmButton: false,

            timer: 2000

          })

          this.router.navigate(['/home']);

        }

        else{

          Swal.fire({

            icon: 'error',

            title: 'Oops...',

            text: data.responseMessage,

          })

         this.router.navigate(['/login']);

         this.loading = false;

        }

      },

    (*err* : HttpErrorResponse)=>{

      this.isLoginError = true;

    });

    const logindata = new Login();

    this.authService.authenticate(this.isLoginError);

  }

}

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**RegistrationComponent:-**

import { Component, OnInit } from '@angular/core';

import { FormBuilder, FormGroup, Validators } from '@angular/forms';

import { Router } from '@angular/router';

import { User } from '../\_models';

import { RegisterService } from '../register.service';

import Swal from 'sweetalert2';

@Component({

  selector: 'app-register',

  templateUrl: './register.component.html'

})

export class RegisterComponent implements OnInit {

  constructor(

    private *formBuilder*: FormBuilder,

    private *router*: Router,

    private *registerService* : RegisterService,

  ) { }

  identityType = [

       { name: "Aadhar Card", value:"aadhar"},

      { name: "PAN card", value:"pancard"},

      { name: "Passport", value:"passport"},

      { name: "Voter Id Card", value: "voter" }

    ]

  registerForm: FormGroup;

  loading = false;

  submitted = false;

  selectedOption:string;

  ngOnInit() {

    this.registerForm = this.formBuilder.group({

      firstName: ['', Validators.required],

      lastName: ['', Validators.required],

      userName:['',Validators.required],

      password: ['', [Validators.required, Validators.minLength(6)]],

      dob:['',Validators.required],

      phone: ['', Validators.required],

      address:['',Validators.required],

      identityType:['',Validators.required],

      file:['',Validators.required],

      identity:['',Validators.required],

      email:['',[Validators.required,Validators.email]],

  });

  }

  get fval() { return this.registerForm.controls; }

  onFormSubmit(){

    this.submitted = true;

    // return for here if form is invalid

    if (this.registerForm.invalid) {

      return;

    }

    this.loading = true;

        const result:User= Object.assign({}, this.registerForm.value);

        // Do useful stuff with the gathered data

        console.log(result.firstName);

          this.registerService.insertUser(result.firstName, result.lastName,result.userName,result.password,result.dob,result.phone, result.address, this.selectedOption,result.identity,result.email).subscribe(

            (*data* : any) =>{

              this.loading=false;

              localStorage.clear();

              localStorage.setItem('user',JSON.stringify(data));

              if(data.registrationStatus==true){

                Swal.fire(

                  {

                    icon: 'success',

                    title: 'User registered succesfully!',

                    text:"Please wait for an email for account activation!"

                  }

                )

              }else{

                Swal.fire({

                  icon: 'error',

                  title: 'Oops...',

                  text: data.responseMessage,

                })

              }

              this.router.navigate(['/login']);

            }

          );

  }

  filterSelected(*selectedOption*){

    this.selectedOption= selectedOption;

    console.log('selected value= '+selectedOption);

  }

}