public class SendQuotationController {

@AuraEnabled

public static void savepdfQuote(string QuoteId){

PageReference pdf = page.CreatePDF;

pdf.getParameters().put('Id', QuoteId);

system.debug('QuoteId===='+QuoteId);

List<Quote> quoteList = [SELECT Id,Quote\_Number\_\_c,PDF\_Version\_\_c,Name FROM Quote WHERE Id=:QuoteId];

if(QuoteId !=null && QuoteId !=''){

BLOB pdfBlob;

if (!Test.isRunningTest()) {

pdfBlob = pdf.getContent(); //generate the pdf BLOB

} else {

pdfBlob = BLOB.valueOf('Test'); // Provide a dummy Blob in the test context

}

System.debug('quoteList====='+quoteList);

// quoteList[0].PDF\_Version\_\_c = quoteList[0].PDF\_Version\_\_c +1;

// update quoteList;

System.debug('quoteList[0].PDF\_Version\_\_c====='+quoteList[0].PDF\_Version\_\_c);

ContentVersion objCV = new ContentVersion();

objCV.VersionData = pdfBlob;

if(quoteList[0].PDF\_Version\_\_c ==0){

system.debug('111111');

objCV.Title = quoteList[0].Quote\_Number\_\_c+ '.pdf';

objCV.PathOnClient = quoteList[0].Quote\_Number\_\_c+ '.pdf';

}else{

objCV.Title = quoteList[0].Quote\_Number\_\_c +'-R'+ quoteList[0].PDF\_Version\_\_c + '.pdf';

objCV.PathOnClient = quoteList[0].Quote\_Number\_\_c+'-R'+ quoteList[0].PDF\_Version\_\_c + '.pdf';

}

insert objCV;

quoteList[0].PDF\_Version\_\_c = quoteList[0].PDF\_Version\_\_c +1;

update quoteList;

System.debug('objCV ' + objCV);

list<ContentVersion> lstConVer = [select id, ContentDocumentId, ContentDocument.id, ContentDocument.title from ContentVersion where id =: objCV.Id limit 1];

ContentDocumentLink objCVN = new ContentDocumentLink();

objCVN.ContentDocumentId = lstConVer[0].ContentDocumentId;

objCVN.LinkedEntityId = QuoteId;

objCVN.ShareType = 'V';

objCVN.Visibility = 'AllUsers';

insert objCVN;

}

}

@AuraEnabled

public static string getContact(String RecordId){

List<Quote> enqList = [SELECT Id,name,ContactId,Email, Opportunity.ContactId,ownerId FROM Quote WHERE Id=:RecordId];

system.debug('conId '+enqList[0].ContactId);

list<contact> lstOfCont=new list<contact>();

if(enqList[0].ContactId !=null)

{

lstOfCont=[select email from contact where id=:enqList[0].ContactId];

}

string toEmail;

if(!lstOfCont.isEmpty()){

toEmail=lstOfCont[0].email;

}

if(toEmail !=null && toEmail !='')

{

if(enqList[0].Email !=null && enqList[0].Email !='')

toEmail +=','+enqList[0].Email;

}else if(enqList[0].Email !=null && enqList[0].Email !='')

{

toEmail =enqList[0].Email;

}

List<Id> userIds = new List<Id>();

for(Quote objquote : enqList){

userIds.add(objquote.ownerId);

}

List<User> userRec = [SELECT Id,Email FROM User WHERE Id=:userIds];

Profile adminProfile = [SELECT Id FROM Profile WHERE Name = 'System Administrator'];

List<User> systemAdmin = [SELECT id,Email FROM User WHERE ProfileId =:adminProfile.id];

string CurrentUserEmail=UserInfo.getUserEmail();

wrapper objwrap=new wrapper();

if(toEmail !=null && toEmail !='')

objwrap.contactEmail = toEmail;

objwrap.recOwnerEmail = userRec[0].Email;

//objwrap.adminEmail = systemAdmin[0].Email;

objwrap.adminEmail = CurrentUserEmail;

return JSON.serialize(objwrap);

}

@AuraEnabled

public static void sendEmail(WrapperEmail wrapper){

if(wrapper!=null){

List<messaging.singleEmailMessage> emailMessages = new List<messaging.singleEmailMessage>();

string to = wrapper.toEmail;

string cc = wrapper.ccEmail;

string fr = wrapper.fromEmail;

String subject = 'Quotation Document';

string enqId;

PageReference pdf = page.CreatePDF;

pdf.getParameters().put('Id', wrapper.recId);

System.debug('enqId===='+wrapper.recId);

List<Quote> quoteList = [SELECT Id,Opportunity.ContactId,Name,ContactId,Quote\_Number\_\_c,PDF\_Version\_\_c,Date\_Of\_Quote\_\_c FROM Quote WHERE Id=:wrapper.recId];

list<contact> lstOfCont=[select email,name from contact where id=:quoteList[0].ContactId];

if(!quoteList.isEmpty()){

Messaging.emailFileAttachment attachment = new Messaging.emailFileAttachment();

attachment.setContentType('application/pdf');

string updateNAme;

if(quoteList[0].Quote\_Number\_\_c !=null)

updateNAme=quoteList[0].Quote\_Number\_\_c.replace('/','-');

system.debug('zzzzz'+updateNAme);

if(quoteList[0].PDF\_Version\_\_c==0)

{

system.debug('111111');

attachment.setFileName(updateNAme +'.pdf');

}else{

system.debug('22222'+quoteList[0].Quote\_Number\_\_c);

attachment.setFileName(updateNAme +'-R'+ quoteList[0].PDF\_Version\_\_c + '.pdf');

}

attachment.setInline(false);

Blob pdfBody;

if (!Test.isRunningTest()) {

pdfBody = pdf.getContentAsPDF();

} else {

// Provide a dummy Blob in the test context

pdfBody = Blob.valueOf('Dummy PDF content');

}

attachment.setBody(pdfBody);

Messaging.SingleEmailMessage email = new Messaging.SingleEmailMessage();

email.setSubject(subject);

string body;

if(!lstOfCont.isEmpty()){

body = 'Hi'+' '+ lstOfCont[0].Name +',' + '<br/><br/>' +

'Please find attached document.<br/><br/>' +

'<br/><br/>' +

'Thanks' + '<br/><br/>'+

'SignetPower Team';

}else{

body = 'Hi'+',' + '<br/><br/>' +

'Please find attached document.<br/><br/>' +

'<br/><br/>' +

'Thanks' + '<br/><br/>'+

'SignetPower Team';

}

string[] toAdd = wrapper.toEmail.split(',');

string[] ccAdd;

system.debug('gfhvh'+wrapper.ccEmail);

if(wrapper.ccEmail !=null && wrapper.ccEmail !=''){

ccAdd = wrapper.ccEmail.split(',');

system.debug('gfhvh===='+ccAdd);

}

//string[] fromAdd = wrapper.fromEmail.split(',');

email.setToAddresses(toAdd);

system.debug('asd'+ccAdd);

if(ccAdd !=null && !ccAdd.isEmpty())

{

email.setCCAddresses(ccAdd);

system.debug('gfhvh====>>>>>>>>>>'+ccAdd);

}

system.debug('PPP======'+ccAdd);

email.setHtmlBody(body);

email.setFileAttachments(new Messaging.EmailFileAttachment[] { attachment } );

Messaging.sendEmail( new Messaging.SingleEmailMessage[] { email } );

Quote qty=new Quote();

qty.id=quoteList[0].id;

qty.Date\_Of\_Quote\_\_c=system.today();

update qty;

}

}

}

public class wrapper{

string contactEmail;

string recOwnerEmail;

string adminEmail;

}

public class WrapperEmail {

@auraenabled

public string recId{get;set;}

@auraenabled

public string toEmail{get;set;}

@AuraEnabled

public string ccEmail{get;set;}

@AuraEnabled

public string fromEmail{get;set;}

}

}