public ActionResult uploadmasterpost()

{

//

var file = Request.Files[Request.Files.GetKey(0)];

if (file != null && file.ContentLength > 0)

{

string failreasion = "";

using (DbContextTransaction dbTran = db.Database.BeginTransaction())

{

try

{

// write the server top folder charts of account to work-----

string path = Path.Combine(Server.MapPath("~/ChartofAccounts"),

Path.GetFileNameWithoutExtension(file.FileName) + Convert.ToString(DateTime.Now.Millisecond) + ".xls");

file.SaveAs(path);

//ViewBag.Message = "File uploaded successfully";

// spire library to work with excel---------

Spire.Xls.Workbook workbookS = new Spire.Xls.Workbook();

workbookS.LoadFromFile(path);

//var a = sheetS.Rows.Count();

//if (System.IO.File.Exists(path))

//{

// System.IO.File.Delete(path);

//}

// Delete from server folder------------// now the excell file is in workbooks class

System.IO.File.Delete(path);

// Delete Data ......................

int comp\_cd = Convert.ToInt32(Session["companyid"]);

var delacc = db.AccountGlMaster.Where(x => x.acccat != "M" && x.CompanyUnitId == comp\_cd).ToList();

foreach (var item in delacc)

{

AccountGlMaster acc = db.AccountGlMaster.Find(item.accslno);

db.AccountGlMaster.Remove(acc);

db.SaveChanges();

}

// 4 means the excel file has 4 sheet//

for (int i = 0; i < 4; i++)

{

Spire.Xls.Worksheet sheetS = workbookS.Worksheets[i];

// inner loop for load the cell value to a variable

for (var row = 2; row <= sheetS.Rows.Count(); row++)

{ //Lopp Start ...................................

// Get Excel Data .................................

string AccountNo = sheetS.Range[row, 2].Value;

string Name = sheetS.Range[row, 3].Value.Trim();

string AccountType = sheetS.Range[row, 4].Value;

string level = sheetS.Range[row, 5].Value;

string TransactionType = sheetS.Range[row, 6].Value;

string AccountCode = sheetS.Range[row, 7].Value;

string AccountControlHead = sheetS.Range[row, 8].Value;

if (Name == "Asset" || Name == "Liability" || Name == "Income" || Name == "Expense" || Name == "" || Name == "Name" || level=="" || TransactionType=="")

{

}

else

{

int maxAccNo = db.AccountGlMaster.Where(x => x.CompanyUnitId == comp\_cd).Max(x => x.accno);

if (db.AccountGlMaster.Where(x => x.CompanyUnitId == comp\_cd && x.accnm.ToLower() == Name.ToLower()).Count() > 0)

{

//Danger("Name exists! Try different", true);

}

else

{

//Get System Data .....................................

string Acctype = "";

int MasterAcc = 0;

if (AccountType == "Asset")

{

Acctype = "D";

MasterAcc = 1;

}

else if (AccountType == "Liability")

{

Acctype = "C";

MasterAcc = 2;

}

else if (AccountType == "Income")

{

Acctype = "C";

MasterAcc = 3;

}

else if (AccountType == "Expense")

{

Acctype = "D";

MasterAcc = 4;

}

string posting = "";

if (TransactionType == "Group")

{

posting = "N";

}

else

{

posting = "P";

}

var conacc = db.AccountGlMaster.Where(x => x.accnm == AccountControlHead.Trim()).SingleOrDefault();

if (conacc == null)

{

int sheetno = i + 1;

failreasion = AccountControlHead + " Not Found in Sheet " + sheetno;

}

// insert data to Database .......................................

AccountGlMaster accountGlMaster = new AccountGlMaster();

accountGlMaster.accno = maxAccNo + 1;

accountGlMaster.CompanyUnitId = comp\_cd;

accountGlMaster.acccat = "C";

accountGlMaster.accnm = Name;

accountGlMaster.opendt = DateTime.Now;

accountGlMaster.acctype = Acctype;

accountGlMaster.acclevel = Convert.ToInt32(level);

accountGlMaster.posting = posting;

accountGlMaster.conacc = conacc.accno;

accountGlMaster.lasttransdt = DateTime.Now;

accountGlMaster.curbal = 0;

accountGlMaster.sysopenbal = 0;

accountGlMaster.accdesc = "";

accountGlMaster.master = MasterAcc;

accountGlMaster.accflag = null;

db.AccountGlMaster.Add(accountGlMaster);

db.SaveChanges();

// End Insert Data .................................................

}

}

//Loop End ............................................

}

} // Sheet loop End .................

ViewBag.Message = "Data Successfully Inserted ! ! !";

dbTran.Commit();

}

catch (Exception e)

{

ViewBag.Message = e.Message + "\n " + failreasion;

dbTran.Rollback();

}

}// Transaction Close .................

}

return Json(ViewBag.Message, JsonRequestBehavior.AllowGet);

}