[HttpPost]

public ActionResult AddProjectTracking([Bind(Include = "CompanyId, TeamId, ProjectMasterId, SectionId, Remarks," +

"ProjectActiveStatus, StartDateStr, EndDateStr, DataCollecting, TargetSetting, SOPformalization, SOPactivation, " +

"ProjectImprovmentBefore, ProjectImprovmentAfter ")] ProjectStatusTracking projectStatusTracking)

{

if (!ModelState.IsValid)

{

return Json(new { error = true, message = "Please input all the information properly." }, JsonRequestBehavior.AllowGet);

}

using (QCCircleProjectDBcontext dBcontext = new QCCircleProjectDBcontext())

{

using (var dbTransaction = dBcontext.Database.BeginTransaction())

{

try

{

projectStatusTracking.StartDate = DateParse.StringToDateTime(projectStatusTracking.StartDateStr);

projectStatusTracking.EndDate = DateParse.StringToDateTime(projectStatusTracking.EndDateStr);

ProjectStatusTrackingGateway projectStatusTrackingGateway = new ProjectStatusTrackingGateway();

int projectId = projectStatusTrackingGateway.AddProjectTracking(projectStatusTracking, dBcontext);

if (projectId <= 0)

{

throw new Exception("Saving failed.");

}

if (Request.Files.Count > 0)

{

for (int i = 0; i < Request.Files.Count; i++)

{

if (Request.Files.AllKeys[i] == "Files")

{

HttpPostedFileBase files = Request.Files[i];

FileAttach\_ProjectStatusTracking fileAttach\_ProjectStatusTracking = new FileAttach\_ProjectStatusTracking();

fileAttach\_ProjectStatusTracking.ProjectTrackingId = projectId;

fileAttach\_ProjectStatusTracking.FileName = "Files";

fileAttach\_ProjectStatusTracking.InsertedBy = MySession.Current.UserNo;

fileAttach\_ProjectStatusTracking.UpdatedBy = MySession.Current.UserNo;

fileAttach\_ProjectStatusTracking = SaveOrderFile(files, fileAttach\_ProjectStatusTracking);

if (fileAttach\_ProjectStatusTracking == null || fileAttach\_ProjectStatusTracking.ProjectTrackingId <= 0)

{

throw new Exception("File saving failed.");

}

int rowAffected = projectStatusTrackingGateway.AddProjectTrackingFile(fileAttach\_ProjectStatusTracking, dBcontext);

if (rowAffected <= 0)

{

throw new Exception("File information saving failed.");

}

}

}

}

dbTransaction.Commit();

return Json(new { error = false, message = "Saved Successfully." }, JsonRequestBehavior.AllowGet);

}

catch (Exception ex)

{

dbTransaction.Rollback();

return Json(new { error = true, message = "Oops! Project tracking is not saved. " + ex.Message }, JsonRequestBehavior.AllowGet);

}

}

}

}

public FileAttach\_ProjectStatusTracking SaveOrderFile(HttpPostedFileBase file, FileAttach\_ProjectStatusTracking fileAttach\_ProjectStatusTracking)

{

using (QCCircleProjectDBcontext dBcontext = new QCCircleProjectDBcontext())

{

dBcontext.Database.CommandTimeout = 1000;

try

{

BaseClassUtility baseClassUtility = new BaseClassUtility();

//FileAttach\_ProjectStatusTracking result = new FileAttach\_ProjectStatusTracking();

fileAttach\_ProjectStatusTracking.FileName = "File\_" + baseClassUtility.RandomString(6) + Path.GetExtension(file.FileName);

string directoryPath = Server.MapPath("/WorkStations/Assets/Files");

if (!Directory.Exists(directoryPath))

{

Directory.CreateDirectory(directoryPath);

}

string serverPath = Path.Combine(Server.MapPath("/WorkStations/Assets/Files"), fileAttach\_ProjectStatusTracking.FileName);

string location = "/WorkStations/Assets/Files/" + fileAttach\_ProjectStatusTracking.FileName;

location = fileAttach\_ProjectStatusTracking.location;

bool fileExist = System.IO.File.Exists(serverPath);

if (fileExist)

{

throw new Exception();

}

file.SaveAs(serverPath);

ProjectStatusTrackingGateway projectStatusTrackingGateway = new ProjectStatusTrackingGateway();

int generatedId = projectStatusTrackingGateway.AddProjectTrackingFile(fileAttach\_ProjectStatusTracking, dBcontext);

fileAttach\_ProjectStatusTracking.Id = generatedId;

return fileAttach\_ProjectStatusTracking;

}

catch (Exception ex)

{

return null;

}

}

}