# آموزش اتصال به درگاه بانکی از طریق ASP.Net MVC

در این مقاله سعی داریم اتصال به درگاه پرداخت اینترنتی بانک ملت را توسط Asp.Net MVC به شما آموزش دهیم

بعد از ثبت نام شما در وب سایت بانک ملت، مقاله ای از طرف بانک برای شما ارسال میشود که پیشنهاد میکنیم حتما آن را مطالعه بفرمایید.

در اولین مرحله به سراغ جدول درگاه پرداخت رفته و به طراحی آن می‌پردازیم

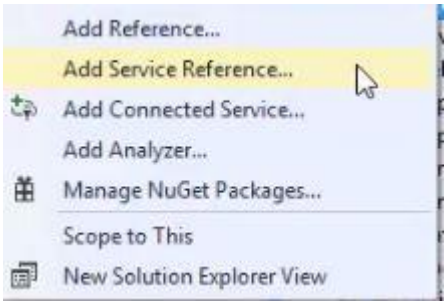
یک جدول با نام Payment طراحی و فیلد های زیر را به جدول اضافه می کنیم.

| | Column Name | Data Type | Allow Nulls |
|---|---|---|---|
| 🔑 | PaymentId | int | ☐ |
| | ReferenceNumber | nvarchar(100) | ☑ |
| | SaleReferenceId | bigint | ☐ |
| | StatusPayment | nvarchar(100) | ☑ |
| | PaymentFinished | bit | ☐ |
| | Amount | bigint | ☐ |
| | BankName | nvarchar(50) | ☑ |
| | UserID | int | ☐ |
| | BuyDatetime | datetime | ☐ |

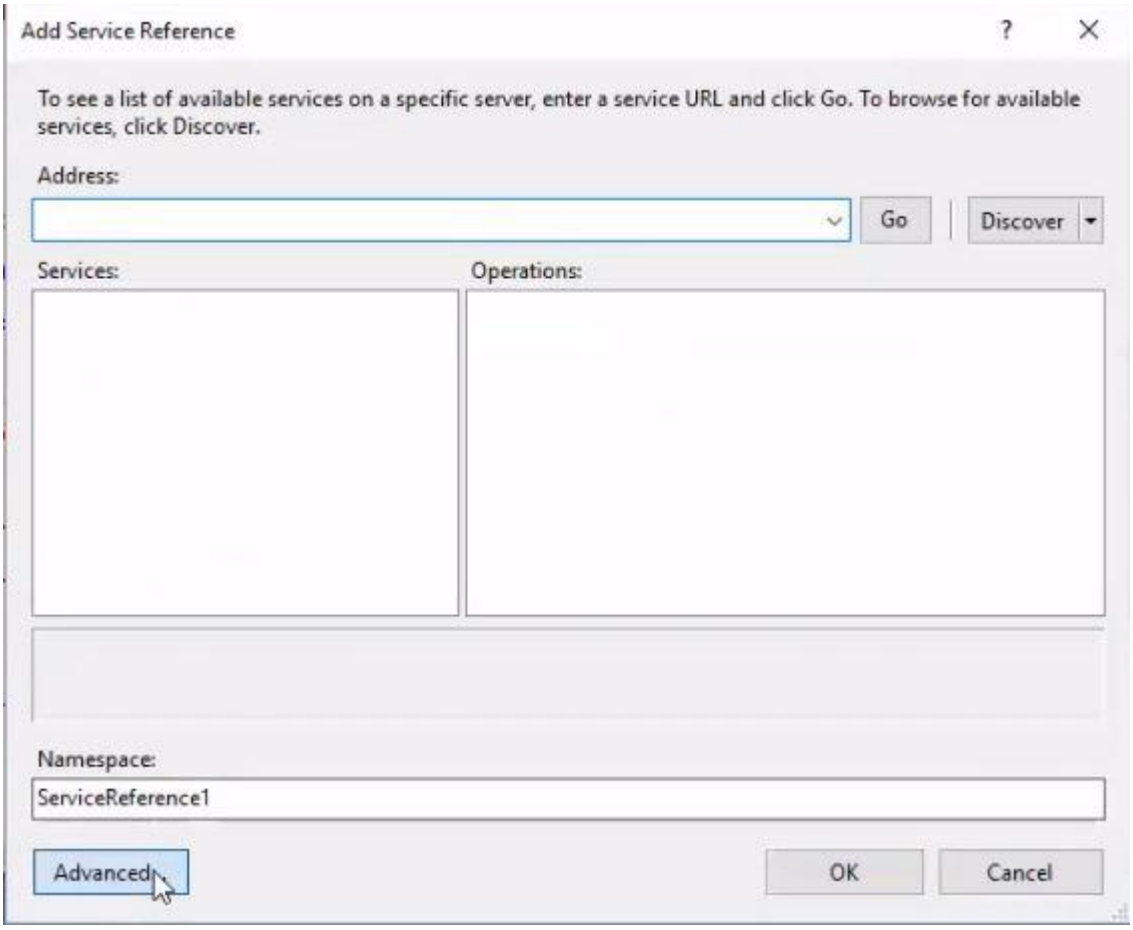
- PaymentId :شماره رکورد پرداخت است که توسط خود دیتابیس مقدار دهی می شود.
- ReferenceNumber: شماره پیگیری فاکتور موجود.
- SaleReferenceId : شماره پرداخت فاکتور موجود.
- StatusPayment: وضعیت پرداخت.
- PaymentFinished: وضعیت نهایی پرداخت.
- Amount: مبلغ پرداخت شده.
- BankName: نام بانک سرویس دهنده.
- UserID: کد کاربر خریدار.
- BuyDatetime: تاریخ خرید.

بعد از طراحی جدول در دیتابیس، وارد Visual Studio میشویم و یک پروژه از نوع Asp.Net Mvc ایجاد می کنیم.

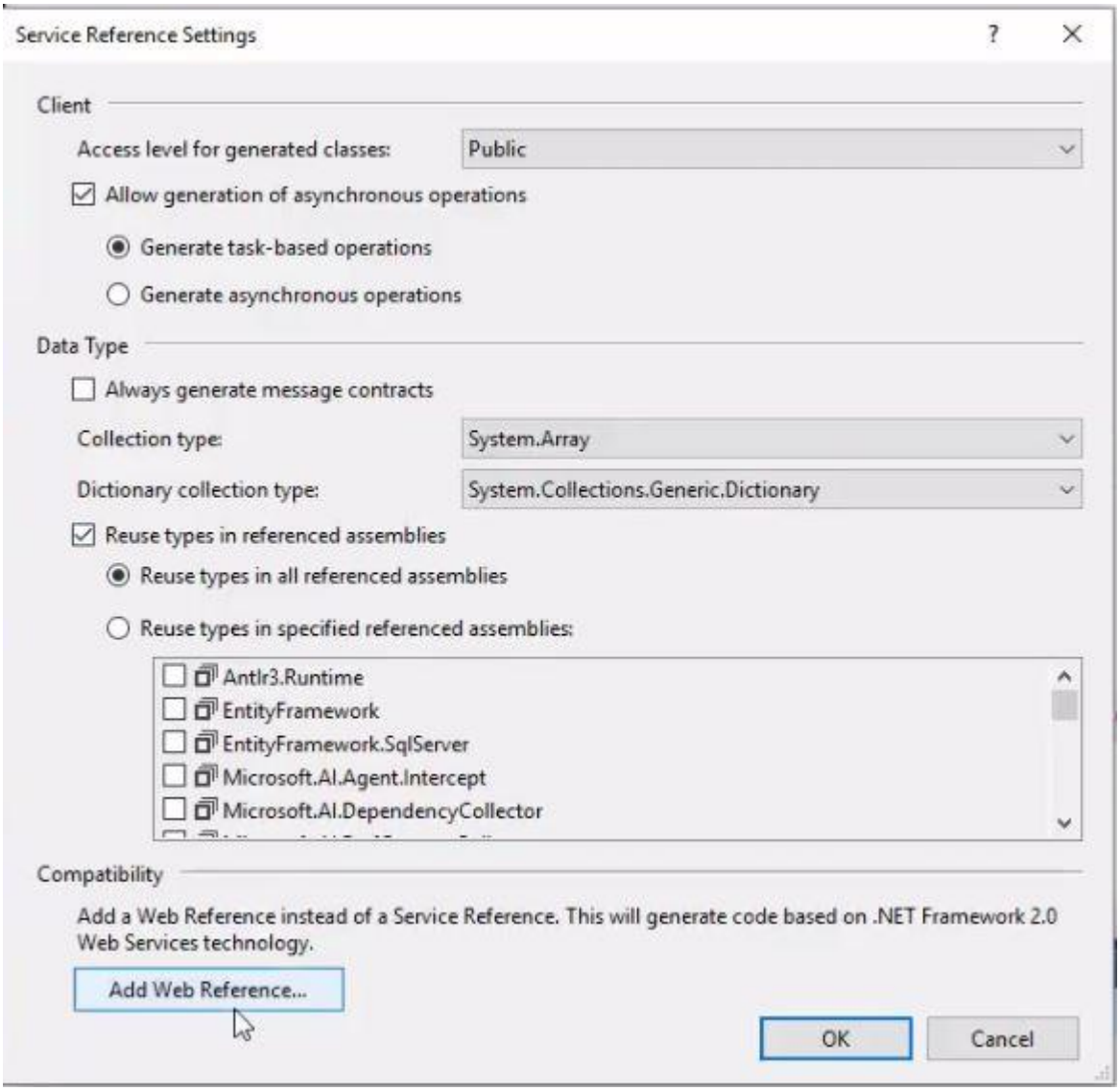
بعد از ایجاد پروژه، به قسمت Refrence ها رفته و بر روی Refrence کلیک راست کرده و گزینه Add Service Refrence را انتخاب می کنیم




**وارد پنجره زیر می شویم و گزینه Advance را کلیک می کنیم**

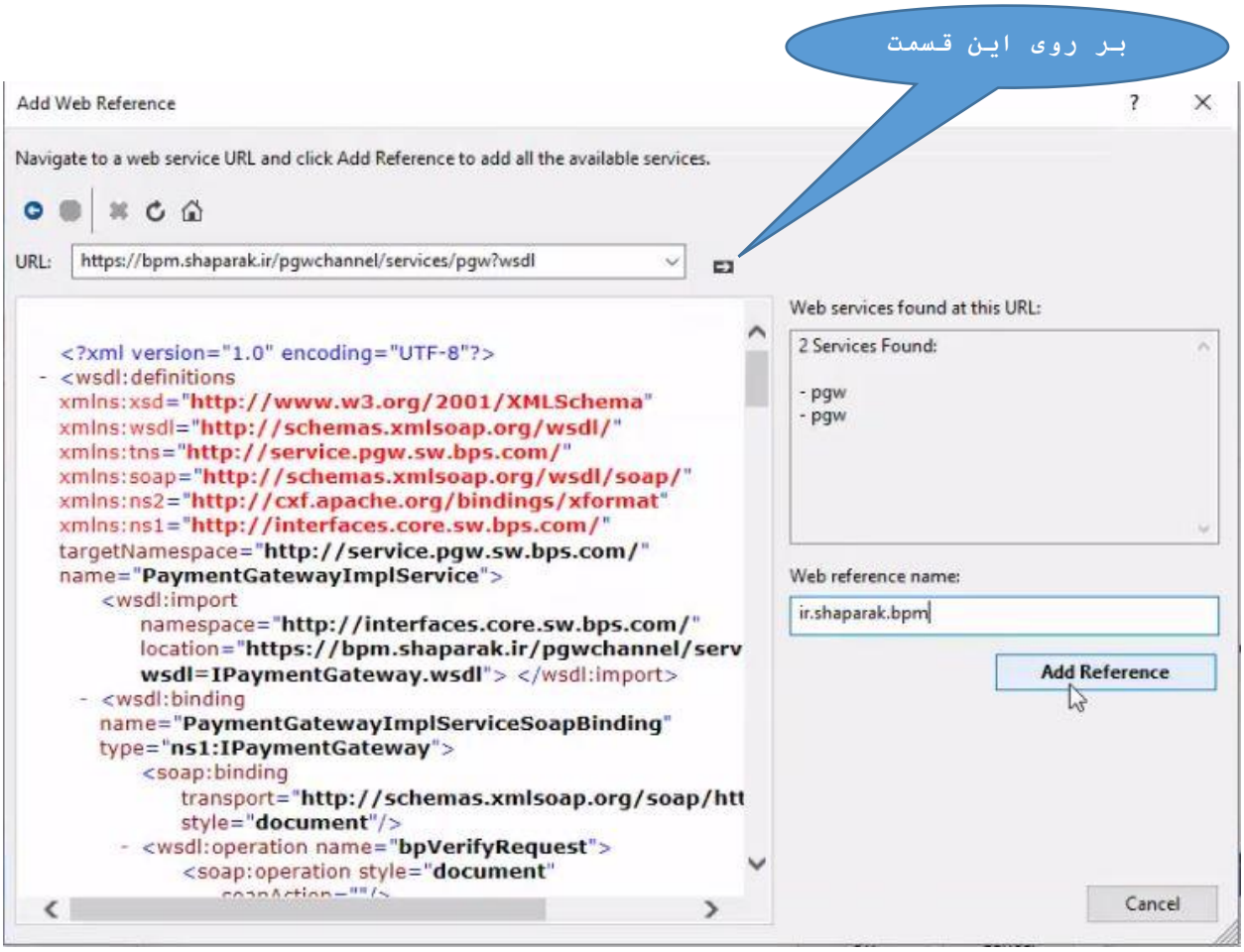



**در پنجره زیر AddWeb Service را کلیک می کنیم.**




**وارد پنجره زیر که شدیم از URL زیر یک کپی تهیه می کنیم و در قسمت URL جایگذاری کرده و گزینه روبرو را کلیک کرده تا وب سرویس آماده شود**

https://bpm.shaparak.ir/pgwchannel/services/pgw?wsdl




**بعد از Load شدن سرویس ، می‌توانید نام سرویس را تغییر داده و بر روی Add Reference کلیک کنید.**

**بعد از اضافه کردن Reference ، یک پوشه به نام Web References به پروژه اضافه می گردد.**

**در مرحله بعد بانک موجود را به روش Data base First به پروژه اضافه می کنیم.**

**حال به سراغ Controller ها رفته و یک کنترلر با نام دلخواه به پروژه اضافه می کنیم.**

**در کنترلر جاری، یک Action با نام Index اضافه می کنیم و در View آن کد های زیر را جایگذاری می کنیم**

```
@{
    Layout = null;
}

<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <link href="~/Content/bootstrap.css" rel="stylesheet" />
    <link href="~/Content/Site.css" rel="stylesheet" />
    <title>پرداخت فرم</title>
</head>
<body dir="rtl">
@using (Html.BeginForm())
```

```
{
    @Html.AntiForgeryToken()

    <div class="container">
        <div class="row">
            <section class="col-lg-4"></section>
            <div class="col-lg-4 border">
                <h1 class="text-center">داده تحلیل آموزشگاه</h1>

                <table class="table table-responsive ">
                    <tr>
                        <td>مبلغ</td>
                        <td>
                            @Html.TextBox("Price", null, new {@class = "form-control"})
                        </td>
                    </tr>
                    <tr>
                        <td colspan="2">
                            <span>بانک درگاه انتخاب : </span>
                        </td>
                    </tr>
                    <tr>

                        <td>
                            @Html.RadioButton("Dargah", "ملت بانک", true, new {@class =
"radio"})
                            <section>ملت  بانک</section>

                        </td>
                        <td >
                            <input type="submit" value="پرداخت" class="btn  btn-success
btn-block"/>
                        </td>
                    </tr>
                    <tr>
                        <td colspan="2">
                            @ViewBag.Message
                        </td>
                    </tr>
                </table>
            </div>

            <section class="col-lg-4"></section>
        </div>
    </div>
}
</body>
</html>
```

**نتیجه کد های فوق ، یک صفحه Html ساده به ما میدهد که بصورت زیر است**


**یک فرم ساده طراحی کرده ایم که مبلغ و نام بانک پذیرنده را از ما دریافت می کند و به اطلاعات را به اکشن مورد نظر ارسال می کند.چون معمولا مبالغ توسط سامانه و بر اساس قسمت کالا و یا ... انتخاب می شود؛ شما می توانید مبلغ را توسط Session به اکشن مذکور ارسال کنید.**

**بعد از وارد کردن مبلغ، فرم به اکشن همنام پست می شود و در آن اکشن کد های زیر را می نویسیم**

```
[HttpPost]
        public ActionResult Index(long Price, string Dargah)
        {

            string RedirectPage = Url.Action("Success", "Home");
            try
            {
                Peyment p = new Peyment
                {
                    Amount = Price,
                    StatusPayment = "-100",
                    BankName = Dargah,
                    SaleReferenceId = 0,
                    BuyDatetime = System.DateTime.Now,
                    UserID = 1,
                    PaymentFinished = false,
                };
                db.Peyments.Add(p);
                db.SaveChanges();
                int paymentid = p.PaymentId;
                if (paymentid > 0)
                {
                    switch (Dargah)
                    {
                        case "ملت بانک":
                            BypassCertificateError();
                            string TerminalID = "1111";
                            string UserName = "2121";
                            string Password = "0000";
```

```
                                var payment = new
        ir.shaparak.bpm.PaymentGatewayImplService();
                            string result = payment.bpPayRequest(
                                Int64.Parse(TerminalID),
                                UserName,
                               Password,
                                paymentid,
                                Price,
                                GetDate(),
                                GetTime(),
                                "داده تحلیل سایت از خرید",
                                RedirectPage,
                               Int64.Parse("0")

                            );
                            if (result != null)
                            {
                                // 45zm24554654,0
                                string[] ResultArray = result.Split(',');
                                if (ResultArray[0].ToString() == "0")
                                {
                                    UpdatePayment(paymentid, "-100", 0, ResultArray[1],
false);
                                    NameValueCollection collection = new
NameValueCollection();
                                    collection.Add("RefId", ResultArray[1]);

Response.Write(Helper.PreparePOSTForm("https://bpm.shaparak.ir/pgwchannel/startpay.mellat
", collection));

                                }
                                else
                                {
                                    UpdatePayment(paymentid, ResultArray[0].ToString(),
0, null, false);
                                    ViewBag.Message =
PaymentResult.MellatResult(ResultArray[0]);
                                }

                            }
                            else
                            {
                                ViewBag.Message = "ندارد وجود بانک درگاه به اتصال امکان";
                            }

                            break;
                        default:
                            ViewBag.Message = "نشده انتخاب درگاهی هیچ";
                            break;
                    }
                }
            }
            catch (Exception e)
            {
                Console.WriteLine(e);
                throw;
            }
            return View();
        }
```

**در کد های فوق توابعی وجود دارد که آن توابع بصورت تفکیکی در ذیل آمده است**

**این تابع برای رفع خطاهای بانکی و توسط خود بانک ساخته شده که ما در پروژه از آن استفاده می کنیم**

```
void BypassCertificateError()
        {
            ServicePointManager.ServerCertificateValidationCallback +=
                delegate (
                    Object sender1,
                    X509Certificate certificate,
                    X509Chain chain,
                    SslPolicyErrors sslPolicyErrors)
                {
                    return true;
                };
        }
```

**متد زیر برای updateدیتابیس استفاده می شود**

```
private void UpdatePayment(long PaymentID, string VResult, long SaleRefrenceID, string
RefID,
            bool PeymentFinished = true)
        {
            var p = db.Peyments.Find(PaymentID);
            if (PaymentID != null)
            {
                p.StatusPayment = VResult;
                p.SaleReferenceId = SaleRefrenceID;
                p.PaymentFinished = PeymentFinished;
                if (RefID != null)
                {
                    p.ReferenceNumber = RefID;
                }
                db.Entry(p).State = EntityState.Modified;
                db.SaveChanges();
            }
        }
```

**متد های زیر برای تبدیل کردن تاریخ و زمان بصورت رشته ای استفاده می شود**

```
private string GetDate()
        {
            return DateTime.Now.Year.ToString() +
DateTime.Now.Month.ToString().PadLeft(2, '0') +
                   DateTime.Now.Day.ToString().PadLeft(2, '0');
        }




        private string GetTime()
        {
            return DateTime.Now.Hour.ToString().PadLeft(2, '0') +
DateTime.Now.Minute.ToString().PadLeft(2, '0') +
                   DateTime.Now.Second.ToString().PadLeft(2, '0');
        }
```

**تابع PaymentResult انواع کدهای خطای صادر شده از بانک را از ورودی گرفته و رشته خطای معادل آن را به کاربر نمایش می دهد که بصورت یک کلاس نوشته شده و تابع MellatResult درون آن نوشته می شود.**

```
public class PaymentResult
  {
      public static string MellatResult(string ID)
      {
          string result = "";
          switch (ID)
          {
              case "-100":
                  result = "شده لغو پرداخت";
                  break;
              case "0":
                  result = "شد انجام موفقیت با تراکنش";
                  break;

              case "11":
                  result = " است نامعتبر کارت شماره ";
                  break;
              case "12":
                  result = "نيست كافي موجودي ";
                  break;
              case "13":
                  result = " است نادرست رمز ";
                  break;
              case "14":
                  result = " است مجاز حد از بيش رمز كردن وارد دفعات تعداد ";
                  break;
              case "15":
                  result = " است نامعتبر كارت ";
                  break;
              case "16":
                  result = " است مجاز حد از بيش وجه برداشت دفعات ";
                  break;
              case "17":
                  result = " است شده منصرف تراكنش انجام از كاربر ";
                  break;
              case "18":
                  result = " است گذشته كارت انقضاي تاريخ ";
                  break;
              case "19":
                  result = " است مجاز حد از بيش وجه برداشت مبلغ ";
                  break;
              case "111":
                  result = " است نامعتبر كارت كننده صادر ";
                  break;
              case "112":
                  result = " كارت كننده صادر سوييچ خطاي ";
                  break;
              case "113":
                  result = " نشد دريافت كارت كننده صادر از پاسخي ";
                  break;
              case "114":
                  result = "نيست تراكنش اين انجام به مجاز كارت دارنده";
                  break;
              case "21":
```

```
        result = "است نامعتبر پذيرنده ";
        break;
    case "23":
        result = "است داده رخ امنيتي خطاي ";
        break;
    case "24":
        result = "است نامعتبر پذيرنده كاربري اطلاعات ";
        break;
    case "25":
        result = "است نامعتبر مبلغ ";
        break;
    case "31":
        result = "است نامعتبر پاسخ ";
        break;

    case "32":
        result = "باشد نمي صحيح شده وارد اطلاعات فرمت ";
        break;
    case "33":
        result = "است نامعتبر حساب ";
        break;
    case "34":
        result = "سيستمي خطاي ";
        break;
    case "35":
        result = "است نامعتبر تاريخ ";
        break;
    case "41":
        result = "كنيد تلاش دوباره ، است تكراري درخواست شماره";
        break;
    case "42":
        result = "تراكنش Sale نشد يافت";
        break;
    case "43":
        result = "درخواست قبلا Verify است شده داده";
        break;
    case "44":
        result = "درخواست Verfiy نشد يافت";
        break;
    case "45":
        result = "تراكنش Settle است شده";
        break;
    case "46":
        result = "تراكنش Settle است نشده";
        break;
    case "47":
        result = "تراكنش Settle نشد يافت";
        break;
    case "48":
        result = "تراكنش Reverse است شده";
        break;
    case "49":
        result = "تراكنش Refund نشد يافت";
        break;
    case "412":
        result = "است نادرست قبض شناسه ";
        break;
    case "413":
        result = "است نادرست پرداخت شناسه ";
        break;
```

```
                case "414":
                    result = "است نامعتبر قبض کننده صادر سازمان ";
                    break;
                case "415":
                    result = "است رسیده پایان به کاري جلسه زمان ";
                    break;
                case "416":
                    result = "اطلاعات ثبت در خطا ";
                    break;
                case "417":
                    result = "است نامعتبر کننده پرداخت شناسه ";
                    break;

                case "418":
                    result = "مشتري اطلاعات تعریف در اشکال ";
                    break;
                case "419":
                    result = "است گذشته مجاز حد از اطلاعات ورود دفعات تعداد ";
                    break;
                case "421":
                    result = "است نامعتبر  IP";
                    break;
                case "51":
                    result = "است تکراري تراکنش ";
                    break;
                case "54":
                    result = "نیست موجود مرجع تراکنش ";
                    break;
                case "55":
                    result = "است نامعتبر تراکنش ";
                    break;
                case "61":
                    result = "واریز در خطا ";
                    break;

                default:
                    result = string.Empty;
                    break;

            }
            return result;
        }
    }
```

**تابع فوق ، انواع خطاهای بانک ملت می باشد که هر کد نشانه یک خطا در انجام عملیات بانکی می باشد و به عنوان مثال اگر خروجی کد صفر باشد یعنی عملیات با موفقیت انجام شده است.**

**متد Helper یک فرم تولید کرده و به بانک پست می کند که یک url می گیرد و Data که در کد های بالا نوشته شده است .**

```
public class Helper
  {
      public static String PreparePOSTForm(string url, NameValueCollection data)
      {
```

```
            string formID = "PostForm";
                        StringBuilder strForm = new StringBuilder();
            strForm.Append("<form id=\"" + formID + "\" name=\"" + formID + "\"
action=\"" + url + "\" method=\"POST\">");
            foreach (string key in data)
            {
                strForm.Append("<input type=\"hidden\" name=\"" + key + "\" value=\"" +
data[key] + "\">");
            }
            strForm.Append("</form>");
            StringBuilder strScript = new StringBuilder();
            strScript.Append("<script language='javascript'>");
            strScript.Append("var v" + formID + " = document." + formID + ";");
            strScript.Append("v" + formID + ".submit();");
            strScript.Append("</script>");
            return strForm.ToString() + strScript.ToString();
        }
    }
```

**اگر این نتیجه این متد موفقیت آمیز بود ، یعنی کاربر به بانک متصل شده است و در صورت تایید و یا عدم تایید، به اکشن Success رفته که کد های آن به صورت زیر است.**

```
  public ActionResult Success()
        {
            ViewBag.Bank = "ملت";
            MellatReturn();
            return View();
        }
```

**مهمترین متد این پروژه می باشد که بصورت زیر نوشته می شود.**

**متد MellatReturn():**

```
private void MellatReturn()
        {

            BypassCertificateError();

            if (string.IsNullOrEmpty(Request.Params["SaleReferenceId"]))
            {
                //ResCode=StatusPayment
                if (!string.IsNullOrEmpty(Request.Params["ResCode"]))
                {
                    ViewBag.Message =
PaymentResult.MellatResult(Request.Params["ResCode"]);
                    ViewBag.SaleReferenceId = "**************";

                }
                else
                {
                    ViewBag.Message = "نیست قبول قابل رسید";
                    ViewBag.SaleReferenceId = "**************";

                }
            }
            else
            {
```

```
                try
                {
                    string TerminalId = "1224";
                    string UserName = "3569";
                    string UserPassword = "5465";



                    long SaleOrderId = 0;  //PaymentID
                    long SaleReferenceId = 0;
                    string RefId = null;

                    try
                    {
                        SaleOrderId =
long.Parse(Request.Params["SaleOrderId"].TrimEnd());
                        SaleReferenceId =
long.Parse(Request.Params["SaleReferenceId"].TrimEnd());
                        RefId = Request.Params["RefId"].TrimEnd();
                    }
                    catch (Exception ex)
                    {
                        ViewBag.message = ex + "<br/>" + " در ، آمده بوجود پرداخت در مشکلی:وضعیت
شد خواهد داده برگشت خودکار صورت به مبلغ آن است شده کسر شما بانکی حساب از پرداختی وجه که صورتی ";
                        ViewBag.SaleReferenceId = "**************";
                        return;
                    }

                    var bpService = new ir.shaparak.bpm.PaymentGatewayImplService();

                    string Result;
                    Result = bpService.bpVerifyRequest(long.Parse(TerminalId), UserName,
UserPassword, SaleOrderId, SaleOrderId, SaleReferenceId);

                    if (!string.IsNullOrEmpty(Result))
                    {
                        if (Result == "0")
                        {
                            string IQresult;
                            IQresult = bpService.bpInquiryRequest(long.Parse(TerminalId),
UserName, UserPassword, SaleOrderId, SaleOrderId, SaleReferenceId);

                            if (IQresult == "0")
                            {

                                long paymentID = Convert.ToInt64(SaleOrderId);


                                UpdatePayment(paymentID, Result, SaleReferenceId, RefId,
true);

                                ViewBag.Message = "شد انجام موفقیت با پرداخت.";
                                ViewBag.SaleReferenceId = SaleReferenceId;


                                // نهایی پرداخت
                                string Sresult;

                                // پرداخت تایید
```

```
                                Sresult =
bpService.bpSettleRequest(long.Parse(TerminalId), UserName, UserPassword, SaleOrderId,
SaleOrderId, SaleReferenceId);

                                if (Sresult != null)
                                {
                                    if (Sresult == "0" || Sresult == "45")
                                    {
                                        //است شده ستل و تایید تراکنش
                                    }
                                    else
                                    {
                                        //است نشده ستل ولی شده تایید تراکنش
                                    }
                                }
                                else
                                {

                                    string Rvresult;
                                    //مبلغ دادن برگشت عملیات
                                    Rvresult =
bpService.bpReversalRequest(long.Parse(TerminalId), UserName, UserPassword, SaleOrderId,
SaleOrderId, SaleReferenceId);

                                    ViewBag.Message = "شد داده بازگشت تراکنش";
                                    ViewBag.SaleReferenceId = "**************";



                                    long paymentId = Convert.ToInt64(SaleOrderId);

                                    UpdatePayment(paymentId, IQresult, SaleReferenceId,
RefId, false);
                                }
                            }
                            else
                            {
                                ViewBag.Message = PaymentResult.MellatResult(Result);
                                ViewBag.SaleReferenceId = "**************";

                                long paymentId = Convert.ToInt64(SaleOrderId);


                                UpdatePayment(paymentId, Result, SaleReferenceId, RefId,
false);
                            }
                        }
                        else
                        {
                            ViewBag.Message = "نیست قبول قابل رسید شماره";
                            ViewBag.SaleReferenceId = "**************";
                        }

                    }
                    catch (Exception ex)
                    {
                        string errors = ex.Message;
                        ViewBag.Message = "از پرداختی وجه صورتیکه در ، است آمده وجود به پرداخت در مشکلی
شد خواهد داده برگشت خودکار صورت به مبلغ آن است شده کسر شما بانکی حساب";
```

```
                    ViewBag.SaleReferenceId = "**************";

                }
            }
        }
```

**در صورت تایید و یا عدم تایید به View اکشن Success رفته که کد های آن بصورت زیر است**

```
@{
    Layout = null;
}

<!DOCTYPE html>

<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <link href="~/Content/bootstrap.css" rel="stylesheet" />
<link href="~/Content/Site.css" rel="stylesheet" />
    <title>Success</title>

</head>
<body dir="rtl" >
<div class="container">
    <div class="row">
        <div class="col-lg-4"></div>
        <div class="col-lg-4">
            <table class="table table-hover table-responsive">
                <tr>
                    <td class="alert-success">پرداخت درگاه</td>
                    <td>@ViewBag.Bank</td>
                </tr>
                <tr>
                    <td class="alert-success">پرداخت وضعیت</td>
                    <td>@ViewBag.Message</td>
                </tr>
                <tr>
                    <td class="alert-success">رهگیری کد</td>
                    <td>@ViewBag.SaleReferenceId</td>
                </tr>
            </table>
        </div>
        <div class="col-lg-4"></div>
    </div>
</div>
</body>
</html>
```

**کد های Html فوق ، یک صفحه تولید میکند که با توجه به نتیجه متد MellatReturn نتیجه صفحه متفاوت خواهد بود .**

| | |
|---|---|
| درگاه پرداخت | ملت |
| وضعیت پرداخت | رسید قابل قبول نیست |
| کد رهگیری | ********** |

**برای یادگیری بهتر پیشنهاد می کنیم حتما فیلم آموزشی موجود در پیوست مقاله را مشاهده بفرمایید**

**پروژه ، دیتابیس و فیلم آموزشی مقاله فوق ، در پیوست موجود می باشد**