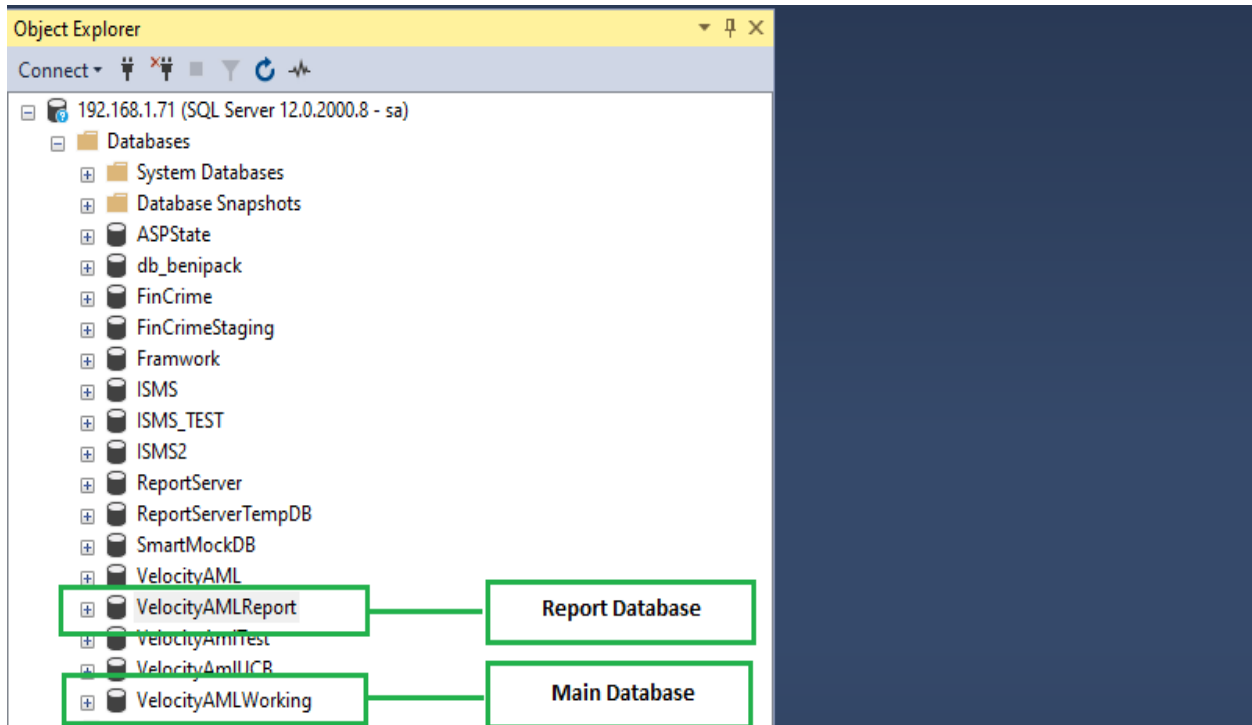
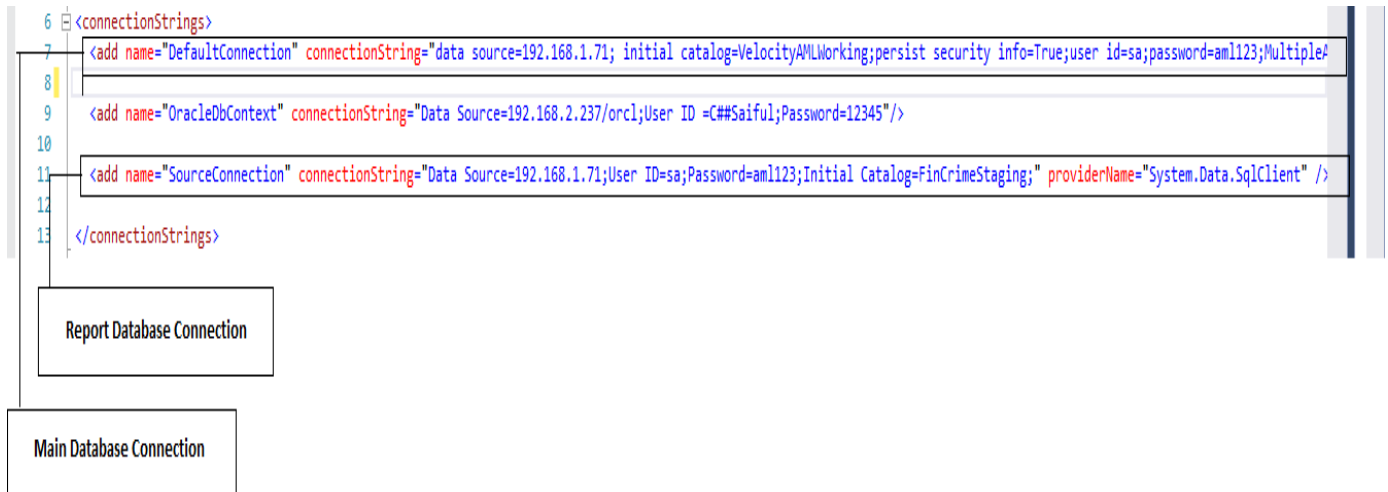


## Separate Report Server Implementation Process in Dot Net Project

**Step 1:** Take main database ( *VelocityAMLWorking* ) backup and restore backup database for report database ( *VelocityAMLReport* ).



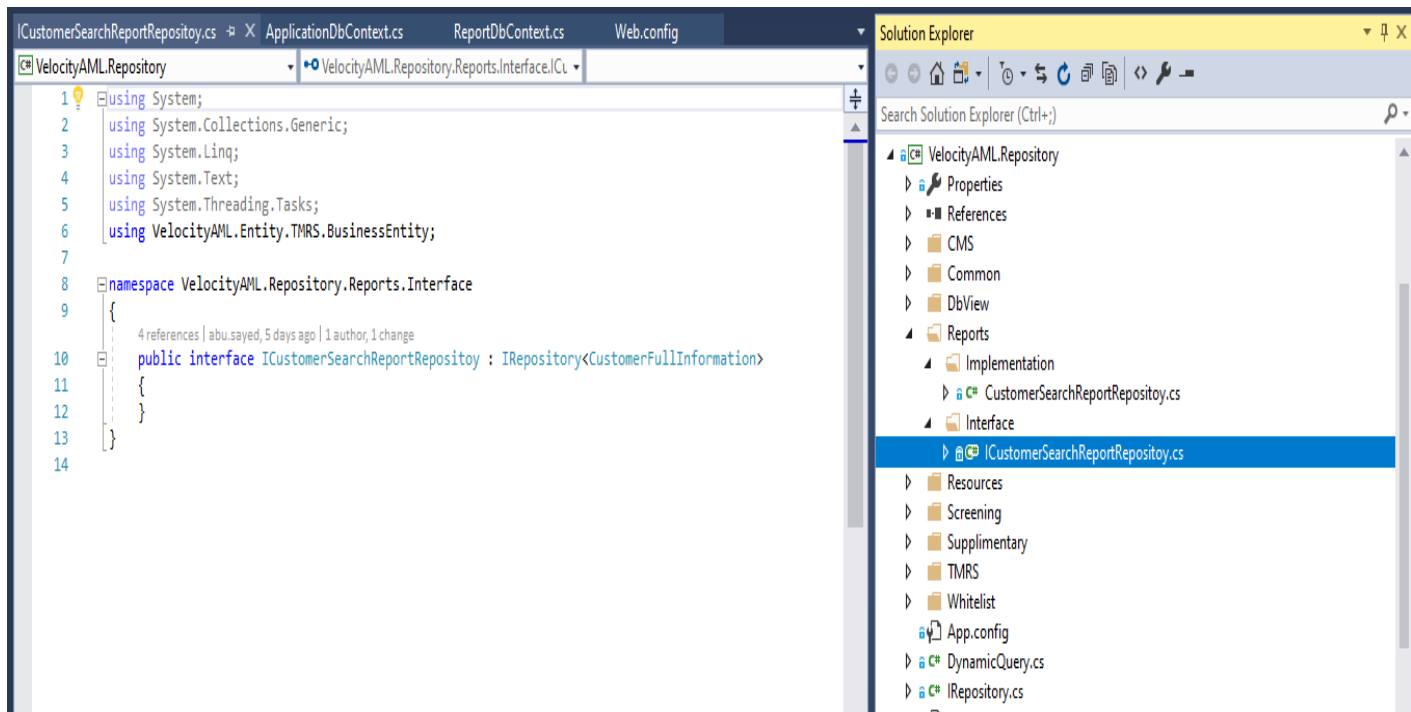
**Step 2:** Add new connectionStrings ( *ReportConnection* ) in connectionStrings section in Webconfig file.



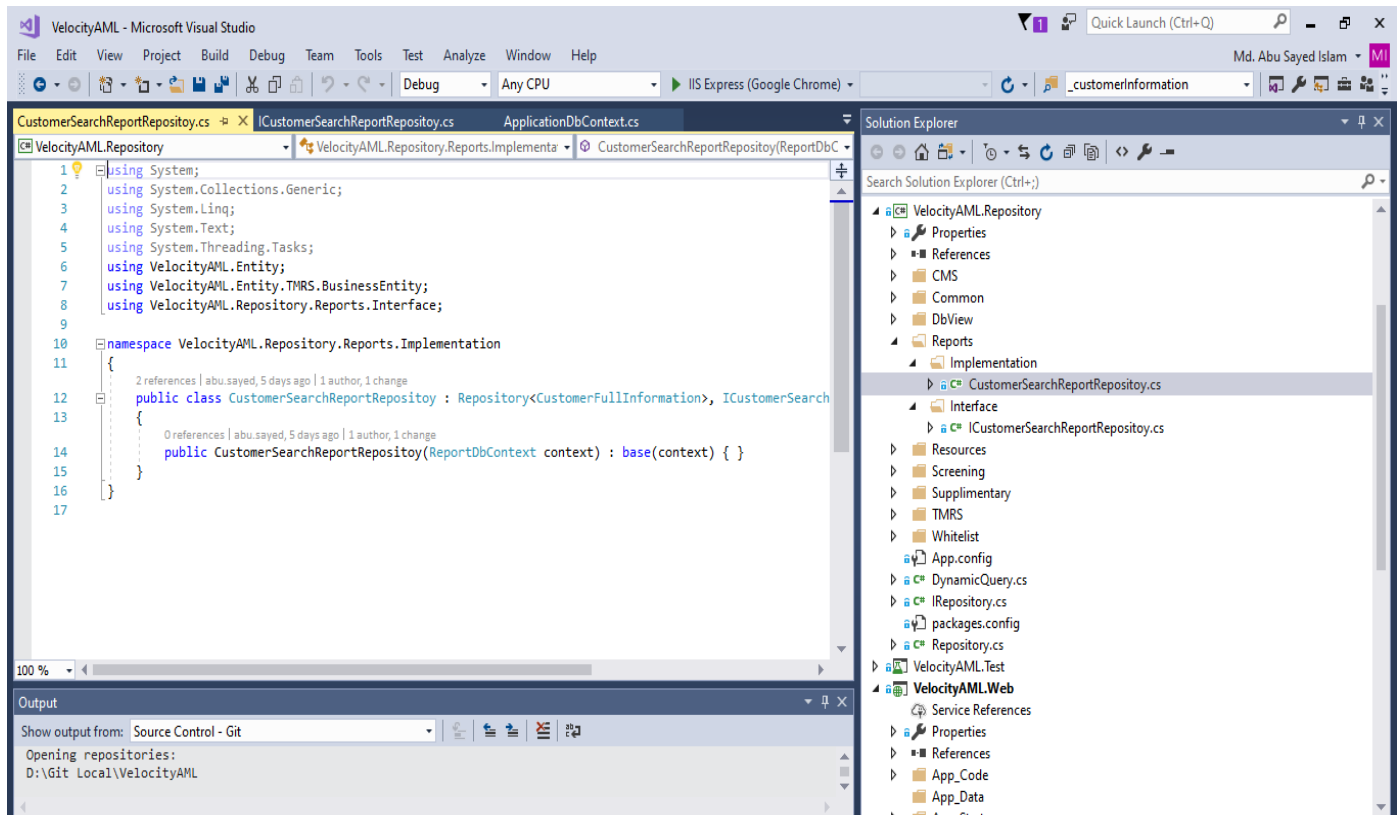
**Step 3:** Add new dbcontext class (*ReportDbContext*) for new report database.



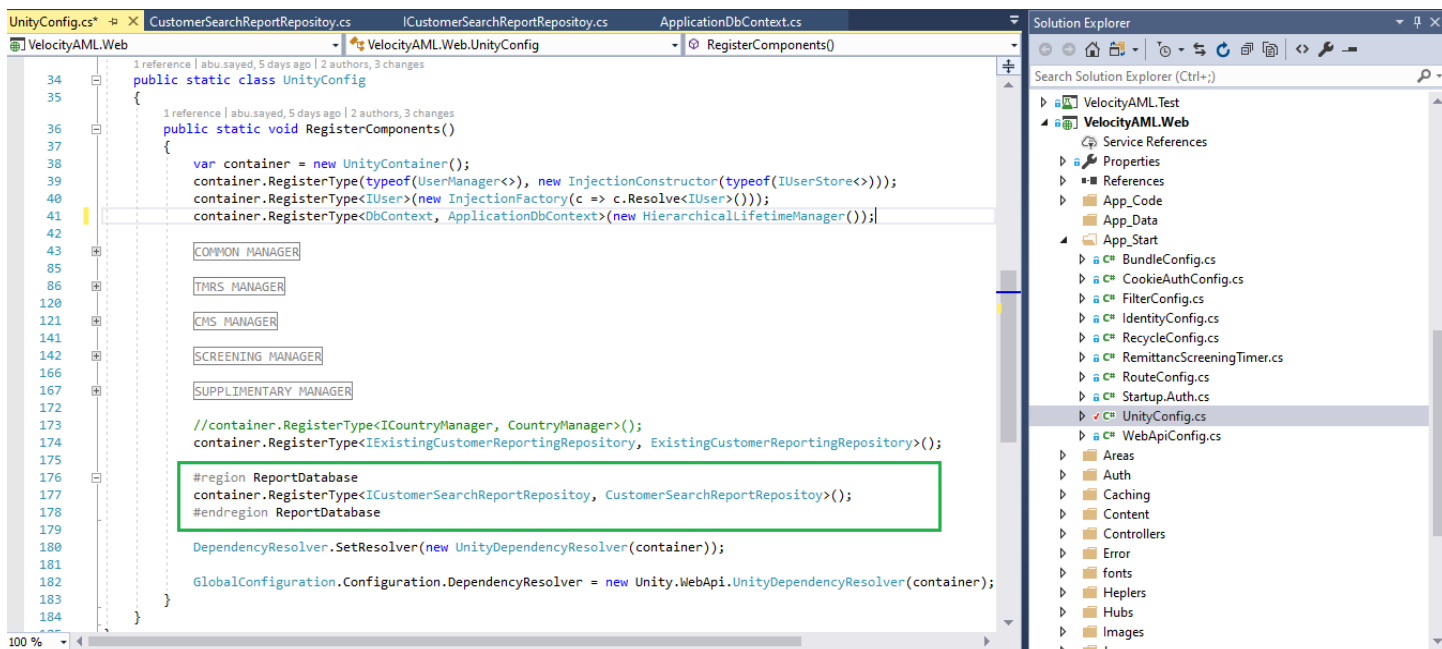
**Step 4:** Add new Interface Repository (*ICustomerSearchReportRepository*)



**Step 5:** Add new Implementation Repository (*CustomerSearchReportRepository*) and pass Dbcontext (*ReportDbContext*) in constructor (*CustomerSearchReportRepository*).



**Step 6:** Register Interface Repository and Implementation Repository in *UnityConfig* class.



**Step 7: Inject Interface Repository (*ICustomerSearchReportRepository*) in Manager Class (*CustomerInformationManager*) and call from it.**

```
33 namespace VelocityAML.TMRS.Manager.Implementation
34 {
35
36     2 references | abu.sayed, 5 days ago | 4 authors, 19 changes
37     public class CustomerInformationManager : ICustomerInformationManager
38     {
39         private readonly ICustomerAccountRepository _customerAccount;
40
41         private readonly ICustomerSearchReportRepository _customerSearchReportRepository;
42         private ISolutionConfigRepository _solution;
43
44         private static readonly Object createLock = new Object();
45         private static readonly Object cusAuthorizeLock = new Object();
46
47         private readonly string _defaultCountry = "";
48
49         0 references | 0 changes | 0 authors, 0 changes
50         public CustomerInformationManager(
51             ICustomerSearchReportRepository customerSearchReportRepository, ISolutionConfigRepository solution)
52         {
53             _customerSearchReportRepository = customerSearchReportRepository;
54             _solution = solution;
55         }
56
57         2450
58         2451
59         2452         IEnumerable<CustomerBasicInfoViewModel> tmodel;
60         2453         if (_solution.GetValue_EmptyIfNotFound("IsUsedReportDatabase", "IsUsedReportDatabase").ToUpper() == "YES")
61         2454         {
62             2455             tmodel = _customerSearchReportRepository.ExecuteQuery<CustomerBasicInfoViewModel>(sql);
63             2456         }
64         2457         else
65         2458         {
66             2459             tmodel = _customerInformation.ExecuteQuery<CustomerBasicInfoViewModel>(sql);
67             2460         }
68         2461
```