Vendor: Microsoft

Exam Code: 070-489

Exam Name: Developing Microsoft SharePoint Server 2013 Advanced Solutions

Version: Demo

Microsoft Exam 070-489 PDF - Developing Microsoft SharePoint Server 2013 Advanced Solutions

Testlet 1

Topic 1, Trey Research

Background

You develop an intranet portal for Trey Research. End users of the portal are researchers and office staff.

Business Requirements

All end users must be able to customize their profile with relevant information. Researchers must store research papers, upload supporting documents, and search content.

Storage

The portal must use an existing Microsoft SQL Server database to access and store work profile information and research papers.

Data Access

- The portal must use Business Connectivity Services (BCS) to access data from external systems.
- Researchers must search content from SharePoint and external systems.
- Researchers must manage a research topic and related content as a single entity.

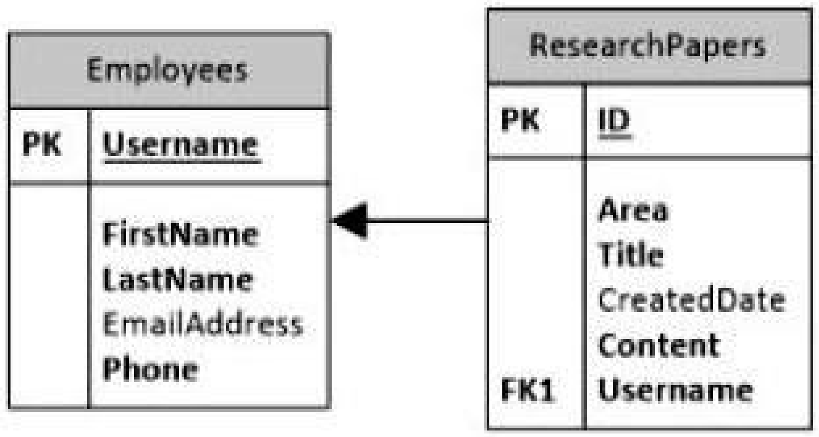
User Profile

- Employees must be able to customize their profile.
- Administrators must be able to create new profile properties.

Technical Requirements

Data Store

The data model for the database entities is shown below:



Users must not be allowed to update the Employees.Username and ResearchPapers.ID fields.
The fields uniquely distinguish the corresponding entity.

Access External Data

- You must create an external content type named TreyResearch to access the SQL data source. During development, the data source will be accessible locally.

- You must develop an app to access the fields named Employee Name and Research Paper Title.
- Researchers must be able to find all research papers that are written by a particular employee.
- A research paper always must be associated with the employee that wrote it.

Document Management

- Researchers must be able to upload research papers and relevant supporting materials into a document set named Research Content. All the document sets must be stored in a list named ResearchPapers. All documents that are uploaded must contain the prefix DOC in the file name.

Environment

The SQL database will be on a different physical server when the solution is deployed to a production environment. The solution must use the SQL Server user named sqltrey to connect to the database. The BCS service is configured and running in the production environment.

Personalize

- You must use custom profile properties to add a new section to the user profile properties page.
- The solution must use the client-side object model (CSOM) to upload employee profile pictures.
- Employees must be able to change their display name on the site. Each employee's page must display the value of the DisplayName and Title fields.

Search

- The Microsoft Bing API web service must be used to search for research papers. No code must be written.
- The app must use a Content Enrichment web service named AbstractIndexer. The app must use the AbstractIndexer service to index search content.
- The solution must store large-sized media files in a dedicated SQL Server database.
- The database must use the ResearchPapers.ID field as the foreign key to associate the field with the TreyResearch external content type.

Application Structure

Relevant portions of the solution files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

App.js

```
AJ01  var context;
AJ02  var web;
AJ03  var user;
AJ04
AJ05
AJ06        $.ajax({
AJ07          url: listURL,
AJ08          headers: {
AJ09            "accept": "application/json",
AJ10            "X-RequestDigest": $("#__REQUESTDIGEST").val()
AJ11          },
AJ12          success: this.showItems,
AJ13          error: this.failMethod
AJ14        });
AJ15       }
AJ16
AJ17       this.showItems = function (data) {
AJ18        $("#Container").children().remove();
AJ19        $.each(data.d.results, function (key, val) {
AJ20          var item = $("#EmployeeInfoTemplate").clone()
AJ21          .attr("id", val.BdcIdentity)
AJ22          .fadeIn("slow");
AJ23           …
AJ24          item.appendTo("#Container");
AJ25        });
AJ26       }
AJ27
AJ28       this.failMethod = function (jqXHR, textStatus, errorThrown) {
AJ29        alert('failed: ' + errorThrown);
AJ30       }
AJ31    }
AJ32   ExecuteOrDelayUntilScriptLoaded(getEmployees, "sp.js");
AJ33  });
AJ34
AJ35  function getEmployees() {
AJ36    var grid = new AppLevelECT.Grid
("ColumnContainer", 3, _spPageContextInfo.webServerRelativeUrl);
AJ37    grid.init();
AJ38  }
```

ManageUserProfiles.es

```
MP01  namespace ManageUserProfiles
MP02  {
MP03    class ProfileProperties
MP04    {
MP05      public static void AddProfileProperty(string name, string displayName,
bool isMultivalued)
MP06      {
MP07        using (SPSite site = new SPSite("http://treyresearch.com/users"))
MP08        {
MP09          SPServiceContext svcContext = SPServiceContext.GetContext(site);
MP10          try
MP11          {
MP12            ProfilePropertyManager prfPropMgr;
MP13            ProfileSubtypeManager prfTypeMgr;
MP14            ProfileSubtypePropertyManager prftypePropMgr;
MP15            ProfileTypePropertyManager typPropMgr;
MP16            ProfileSubtypeProperty prfTypeProp;
MP17            ProfileTypeProperty prfProp;
MP18            ProfileSubtype prfType;
MP19            CorePropertyManager corePropMgr;
MP20            CoreProperty coreProp;
MP21            prfPropMgr = new UserProfileConfigManager(svcContext)
               .ProfilePropertyManager;
MP22
MP23           prfTypeProp = prftypePropMgr.Create(prfProp);
MP24           prfTypeProp.IsUserEditable = true;
MP25           prfTypeProp.DefaultPrivacy = Privacy.Public;
MP26           prfTypeProp.UserOverridePrivacy = true;
MP27           prftypePropMgr.Add(prfTypeProp);
MP28         }
MP29         catch (System.Exception e)
MP30        {
MP31         throw new Exception("Error occurred: " + e.ToString());
MP32        }
MP33       }
MP34     }
MP35   }
MP36 }
MP37
MP38
MP39
MP40 public void UploadPicture(string account, string picURL)
MP41 {
MP42   try
MP43   {
MP44
MP45   }
MP46   catch (Exception e)
MP47   {
MP48     throw new Exception("Error occurred: " + e.ToString());
MP49   }
MP50 }
MP51
MP52 public UserProfileProperties GetUserProfileProperties(string account)
MP53 {
MP54   var userprfProps = new UserProfileProperties();
MP55
MP56   var clientContext = new ClientContext("http://treyresearch.com/users");
```

ContentManagement.es

```
CM01  private void CreateDocumentSets()
CM02  {
CM03    using (SPSite site = new SPSite("http://treyresearch.com/sites"))
CM04    {
CM05      using (SPWeb web = site.RootWeb)
CM06      {
CM07
CM08      }
CM09    }
CM10  }
```

QUESTION 1
You need to configure the external content type to search for research papers.

Which indexing connector should you use?
A. .NET Type Connector
B. WCF Service Connector
C. Custom Connector
D. SQL Server Connector

Correct Answer: B
Explanation

Explanation/Reference:

QUESTION 2
You need to generate document identifiers for each new document that is uploaded to the site.

What should you do?

A. Create a derived class that inherits from the abstract class named Microsoft.Office.DocumentManagement.DocumentId and then override all of the abstract methods.
B. Create a derived class that inherits from the abstract class named Microsoft.Office.DocumentManagement.DocumentIdProvider and then override all of the virtual members.
C. Create a derived class that inherits from the Microsoft.Office.DocumentManagement.DocumentIdProvider abstract class and then implement all abstract members.
D. Create a class to implement the Microsoft.Office.DocumentManagement.IDocumentId interface and then override all of the virtual members.

Correct Answer: B
Explanation

Explanation/Reference:

QUESTION 3
You need to configure authentication to access the SQL data source during development.

Which authentication mechanism should you use?

A. Impersonated Windows Identity
B. Pass Through
C. Impersonated Custom Identity
D. Forms Based Authentication

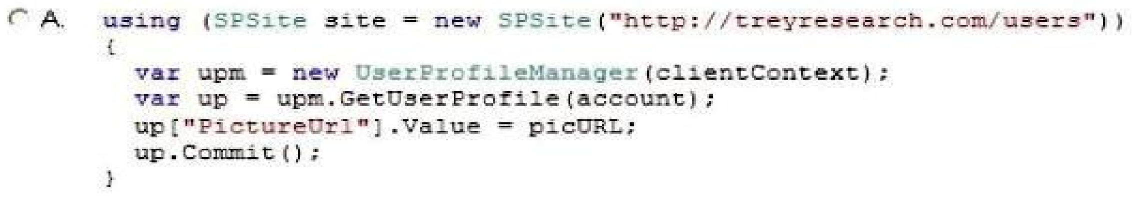
Correct Answer: B
Explanation

Explanation/Reference:

QUESTION 4

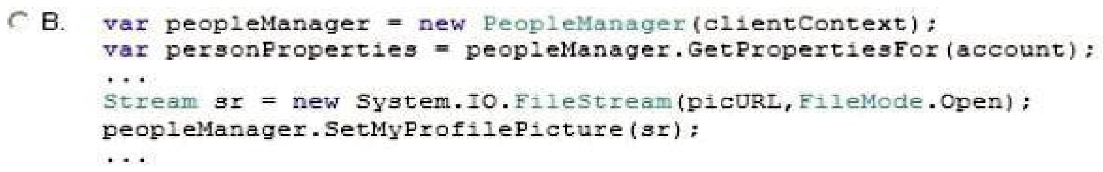
You need to ensure that users can upload pictures.

Which code segment should you insert at line MP57?



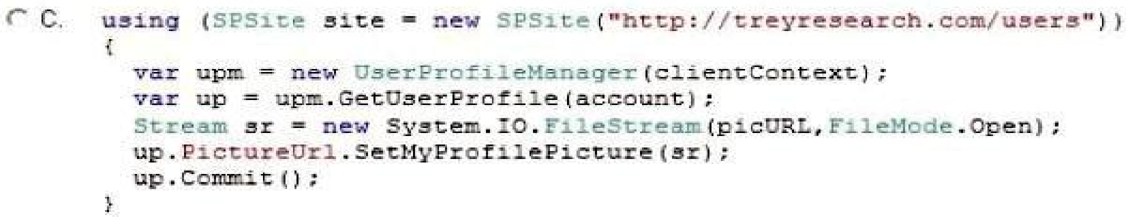
A.

```
using (SPSite site = new SPSite("http://treyresearch.com/users"))
{
  var upm = new UserProfileManager(clientContext);
  var up = upm.GetUserProfile(account);
  up["PictureUrl"].Value = picURL;
  up.Commit();
}
```



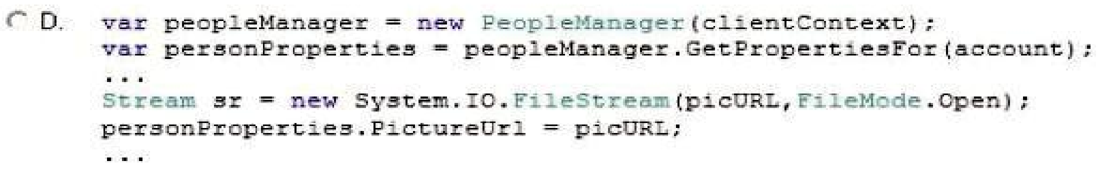
B.

```
var peopleManager = new PeopleManager(clientContext);
var personProperties = peopleManager.GetPropertiesFor(account);
...
Stream sr = new System.IO.FileStream(picURL,FileMode.Open);
peopleManager.SetMyProfilePicture(sr);
...
```



C.

```
using (SPSite site = new SPSite("http://treyresearch.com/users"))
{
  var upm = new UserProfileManager(clientContext);
  var up = upm.GetUserProfile(account);
  Stream sr = new System.IO.FileStream(picURL,FileMode.Open);
  up.PictureUrl.SetMyProfilePicture(sr);
  up.Commit();
}
```



D.

```
var peopleManager = new PeopleManager(clientContext);
var personProperties = peopleManager.GetPropertiesFor(account);
...
Stream sr = new System.IO.FileStream(picURL,FileMode.Open);
personProperties.PictureUrl = picURL;
...
```