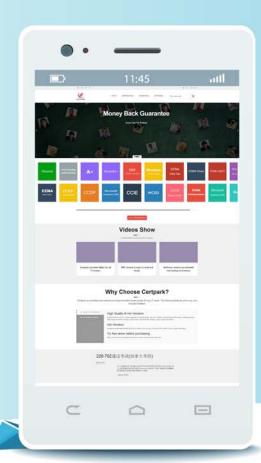
# CERTPARK QUESTION & ANSWER

# **CERTPARK.COM**

Accurate Study Guides,
High Passing Rate!
provides update
free of charge
in one year!



**Exam** : 489

Title : Developing Microsoft

**SharePoint Server 2013** 

**Advanced Solutions** 

Version: DEMO

# 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.

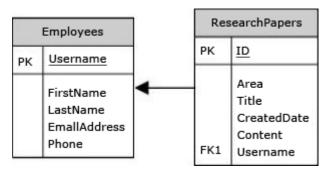
#### **User Profile**

- 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 Research Papers. 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: {
               "accept": "application/json",
AJ09
              "X-RequestDigest": $("# REQUESTDIGEST").val()
AJ10
AJ11
            },
            success: this.showItems,
AJ12
AJ13
            error: this.failMethod
AJ14
          1);
AJ15
         1
AJ16
AJ17
         this.showItems = function (data) {
          $("#Container").children().remove();
AJ18
AJ19
          $.each(data.d.results, function (key, val) {
AJ20
            var item = $("#EmployeeInfoTemplate").clone()
AJ21
            .attr("id", val.BdcIdentity)
AJ22
            .fadeIn("slow");
AJ23
            item.appendTo("#Container");
AJ24
AJ25
          1);
AJ26
         3
AJ27
AJ28
         this.failMethod = function (jqXHR, textStatus, errorThrown) {
          alert('failed: ' + errorThrown);
AJ29
AJ30
AJ31
      1
AJ32
      ExecuteOrDelayUntilScriptLoaded(getEmployees, "sp.js");
AJ33 });
AJ34
AJ35 function getEmployees() {
       var grid = new AppLevelECT.Grid
AJ36
("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;
                ProfileTypePropertyManager typPropMgr;
MP15
                ProfileSubtypeProperty prfTypeProp;
MP16
                ProfileTypeProperty prfProp;
MP17
               ProfileSubtype prfType;
MP18
MP19
               CorePropertyManager corePropMgr;
MP20
               CoreProperty coreProp;
MP21
                prfPropMgr = new UserProfileConfigManager(svcContext)
               .ProfilePropertyManager;
MP22
               prfTypeProp = prftypePropMgr.Create(prfProp);
MP23
MP24
               prfTypeProp.IsUserEditable = true;
              prfTypeProp.DefaultPrivacy = Privacy.Public;
MP25
MP26
              prfTypeProp.UserOverridePrivacy = true;
MP27
               prftypePropMgr.Add(prfTypeProp);
MP28
             catch (System.Exception e)
MP29
MP30
             throw new Exception ("Error occurred: " + e.ToString());
MP31
MP32
MP33
           3
         }
MP34
MP35
MP36 }
MP37
MP38
MP39
MP40 public void UploadPicture(string account, string picURL)
MP41 {
MP42
     try
MP43
     -{
MP44
MP45
     catch (Exception e)
MP47
         throw new Exception ("Error occurred: " + e.ToString());
MP48
MP49
MP50 }
MP51
MP52 public UserProfileProperties GetUserProfileProperties(string account)
MP53 {
MP54
       var userprfProps = new UserProfileProperties();
MP55
     var clientContext = new ClientContext("http://treyresearch.com/users");
MP56
```

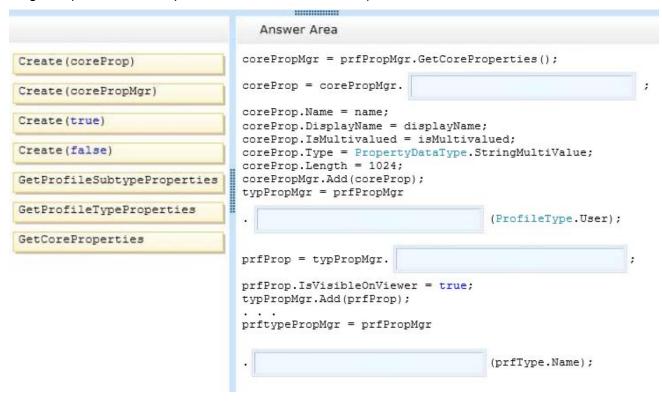
# ContentManagement.es

```
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 }
```

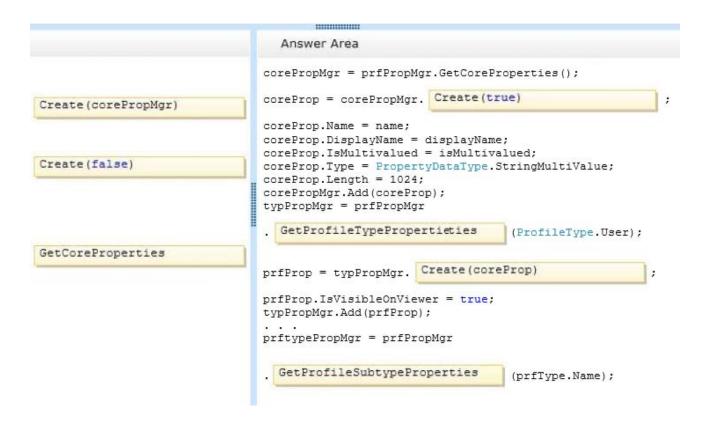
# **DRAG DROP**

You need to add code to line MP22 to create the custom profile property.

How should you complete the relevant code? (To answer, drag the appropriate code segments to the correct locations in the answer area. Each code segment may be used once or not at all. You may need to drag the split bar between panes or scroll to view content.)

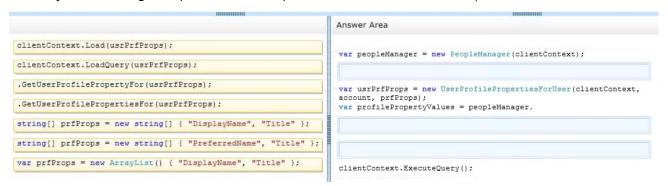


#### Answer:



#### 2.DRAG DROP

You need to add code to line MP57 to display the required properties for the user profile. How should you complete the relevant code? (To answer, drag the appropriate code segments to the correct locations in the answer area. Each code segment may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content.)



#### Answer:

```
Answer Area

var peopleManager = new PeopleManager(clientContext);

string[] prfProps = new string[] { "DisplayName", "Title" };

var prfProps = new string[] { "PreferredName", "Title" };

var prfProps = new ArrayList() { "DisplayName", "Title" };

clientContext.Load(usrPrfProps);

clientContext.Load(usrPrfProps);

clientContext.ExecuteQuery();
```

3. 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

Answer: B

4. 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.

Answer: B

#### 5.DRAG DROP

You need to configure authentication for the external content type in the production environment. Which three actions should you perform in sequence? (To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.)

11111111

#### Answer Area

Add the sqltrey user login to the Secure Store Service.

Configure permissions to allow the user that is logged in to have access to the Secure Store Service.

Stop the Business Data Connectivity service.

Create a Secure Store Service application as a target application.

Connect to the external data source by using the Impersonated Custom Identity and the target application name.

Connect to the external data source by using the Impersonated Windows Identity and the target application name.

Answer:

# Answer Area Stop the Business Data Connectivity service. Configure permissions to allow the Add the sqltrey user login to the user that is logged in to have access Secure Store Service. to the Secure Store Service. Connect to the external data source by using the Impersonated Windows Identity and the target application name. Create a Secure Store Service application as a target application. Connect to the external data source by using the Impersonated Custom Identity and the target application name.

# **Explanation:**

\* From scenario:

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

The solution must use the SQL Server user named sqltrey to connect to the database.