Testpassport Q&A

La meilleure qualité le meilleur service

Exam : **PL-400**

Title : Microsoft Power Platform Developer

Version : DEMO

1.Topic 1, Bellows Sports

Case study
This is a case study. **Case studies are not timed separately. You can use as much exam time as you would like to complete each case.** However, there may be additional case studies and sections on this exam. You must manage your time to ensure that you are able to complete all questions included on this exam in the time provided.

To answer the questions included in a case study, you will need to reference information that is provided in the case study. Case studies might contain exhibits and other resources that provide more information about the scenario that is described in the case study. Each question is independent of the other questions in this case study.

At the end of this case study, a review screen will appear. This screen allows you to review your answers and to make changes before you move to the next section of the exam. After you begin a new section, you cannot return to this section.

To start the case study
To display the first question in this case study, click the **Next** button. Use the buttons in the left pane to explore the content of the case study before you answer the questions. Clicking these buttons displays information such as business requirements, existing environment, and problem statements. If the case study has an **All Information** tab, note that the information displayed is identical to the information displayed on the subsequent tabs. When you are ready to answer a question, click the **Question** button to return to the question.

Background
Bellows Sports is the region’s newest, largest, and most complete sports complex. The company features baseball and soccer fields and two full-size hockey rinks. The complex provides coaching, recreational leagues, a pro shop, and state-of-the art customer and player amenities.

The company is organized into the following divisions:
- Baseball
- Hockey
- Soccer

Bellow Sports runs tournaments several times per year. Each tournament runs six weeks.

Current environment
Requirements
Bellow Sports tracks players and events in Microsoft Excel workbooks and uses email to communicate with players, partners, and prospective customers. The company uses a proprietary cloud-based accounting system.

The company relies on referrals from athletes for new business. Bellows uses a third-party marketing

company to gather feedback and referrals from athletes. The third-party marketing company uploads a Microsoft Excel file containing lists of potential customers and players to the FTP site that Bellows Sports maintains.

Tournaments

Customer information is stored in the Accounts entity. Each tournament record must list the associated sales representative as the tournament owner. When team members create tournament records they must enter the start date for a tournament. The end date of the tournament must be automatically calculated.

Registration form

You must create a form to allow players to register for tournaments.
The registration form must meet the following requirements:

Division	Requirement
Baseball	Capture the age and weight of the player. The height field must not display.
Hockey	Capture the age, height, and weight of the player.
Soccer	Capture the age of the player. The height and weight fields must not display.

- Each division has tournaments that take place in specific locations. Users must be able to select the division for a tournament location.
- Information about upcoming tournaments must be pre-located into the registration form when the registration form loads.
- The form must include a custom button that sends an email confirmation to the player after the player registers.
- The button must not be visible until after the form is saved.

Security

The company identifies the following job roles:

Role	Tasks
Sales representative	These users will enter data into Dynamics 365 Customer Service.
Intern	These users will create Power Apps apps, and connectors, and create Power Automate flows.
Manager	These users will add users, assign security roles, and manage data storage.

You must grant users the minimum permissions required to perform their job tasks.

Data automation

- Customer name must be added to Dynamics 365 Finance automatically after it is entered.
- You must produce a report that details the number of registrations for a day and send the report as a PDF to the management team.
- You must implement mechanisms to handle all code-related errors.
- When a customer record is updated, the system must look up the account number for the customer in

the accounting system.
- Referrals must be imported into the system as soon as they are available.

Issues
Apps
- The captions for the New and Save buttons do not render properly on the form.
- Interns can create apps but cannot interact with their own data.

Portal

The query for all registered users must return the data categorized by division. Queries must return only the Name and Sport fields. Queries return all fields.

The query is as follows:

```
GET [Organization URI]/api/data/v9.1/accounts?
&$orderby=Name, sport
&$filter=sport ne null
```

Solution checker issues
You run solution checker and observe **Plug-in or workflow activity** errors in the following code sets:

Set	Code	Error message
Code set 1	CS101 var columns = new ColumnSet(); CS102 columns.AllColumns = true; CS103 var query = new QueryExpression("account"); CS104 query.ColumnSet = columns; CS105 var results = service.RetrieveMultiple(query);	il-specify-column
Code Set 2	CS201 WebRequest request = WebRequest.Create ("https://www.bellows.com/api/stuff"); CS202 HttpWebResponse response = new HttpWebResponse(); CS203 CS204 respone = request.GetResponse(); CS205 response.Close();	il-turn-off-keepalive

Code
The following code runs when the registration form loads.
You must implement a mechanism to handle errors that occur in the code:

UpdateRecord.js (Line numbers are included for reference only.)

```
UR01
UR02 var data =
UR03      {
UR04        "name" : "Updated Account"
            "creditonhold" : true,
            "description" : "This is an account update",
            "revenue" : 10,000,
            "Division" : 2
          }
UR05 . . .
UR06
```

You need to configure the system to support automation for referrals.
What are two possible ways to achieve the goal? Each correct selection presents a complete solution.
NOTE: Each correct selection is worth one point.
A. Azure Function that uses the Discovery service
B. workflow extension
C. Azure Function that uses a listener
D. Power Automate flow
Answer: B,D

2.You need to add the script to populate event data on the form.
Which code segment should you use?
A. formContext.data.addOnLoad(myFunction)
B. formContext.data.removeOnLoad(myFunction)
C. formContext.data.entity.addOnSave(myFunction)
D. addOnPreProcessStatusChange
E. formContext.data.isValid()
Answer: D
Explanation:
IsValid() gets a boolean value indicating whether a l of the form data is valid. This includes the main entity and any unbound attributes.
Description: true if a l of the form data is valid; false otherwise.
Reference:
https://docs.microsoft.com/en-us/powerapps/developer/model-driven-apps/clientapi/reference/formcontext-data/isvalid

3.You need to handle errors in UpdateRecord.js.
Which code segment should you add at line UR06?
A. catch(error) {
alert("Caught error: " + error.message);}
B. Exception exception = Server.GetLastError() ;
if(exception != null)}

C. catch(exception e){
console.writeline(e)}
D. function (error){
console.log(error.message)}
Answer: A
Explanation:
The catch statement lets you handle the error.
Syntax: catch(err) {
Block of code to handle errors
}
Reference: https://www.w3schools.com/js/js_errors.asp

4.DRAG DROP
You need to address the user interface issues.
What should you do? To answer, drag the appropriate actions to the correct issues. Each action 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. NOTE: Each correct selection is worth one point.

Actions	Requirement	Action
Add `&ribbondebug=true` to the end of the application URL.	Resolve rendering issue for New and Save buttons.	
Export the XML file.	Add email button for registration form.	
Modify the RibbonWSS.xsd file.		
Use Ribbon Workbench.		

Answer:

Actions	Requirement	Action
Add `&ribbondebug=true` to the end of the application URL.	Resolve rendering issue for New and Save buttons.	Add `&ribbondebug=true` to the end of the application URL.
Export the XML file.	Add email button for registration form.	Use Ribbon Workbench.
Modify the RibbonWSS.xsd file.		
Use Ribbon Workbench.		

Explanation:
Box 1: Add &ribbondebug=true to the end of the application URL.

Scenario: The captions for the New and Save buttons do not render properly on the form.
You can use the an in-app tool called the Command Checker to inspect the ribbon component definitions to help us determine why the button is not rendered correctly.
To enable the Command Checker, you must append a parameter &ribbondebug=true to your D365 application URL.
For example:
https://yourorgname.crm.dynamics.com/main.aspx?appid=9ab590fc-d25e-ea11-a81d-000d3ac2b3e6&ribbondebug=true
Box 2: Use the Ribbon Workbench
Adding Buttons to Ribbons
- Download and install Ribbon Workbench.
- Select a suitable ICON for your button.
- Create a solution.
- Edit the button in Ribbon Workbench.
- Publish and test.

5.DRAG DROP
You need to select connectors for the app.
Which types of connectors should you use? To answer, drag the appropriate connectors to the correct requirements. Each connector 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. NOTE: Each correct selection is worth one point.

Connectors	Requirement	Connectors
Create a custom connector.	View full registration records.	
Use an AppSource connector.	View customer names.	
Use a native application function.	View daily registrations.	
Create a connector with a Postman collection.		

Answer:

Connectors	Requirement	Connectors
Create a custom connector.	View full registration records.	Create a custom connector.
Use an AppSource connector.	View customer names.	Use an AppSource connector.
Use a native application function.	View daily registrations.	Use a native application function.
Create a connector with a Postman collection.		

Explanation:
Box 1: Create a custom connector
A custom connector is a wrapper around a REST API (Logic Apps also supports SOAP APIs) that allows Logic Apps, Power Automate, or Power Apps to communicate with that REST or SOAP API.
Box 2: Use an AppSource connector
You can only retrieve the Customer, UnifiedActivity, and Segments entities through the Power Apps connector. Other entities are shown because the underlying connector supports them through triggers in Power Automate.

Scenario: Customer information is stored in the Accounts entity.
Box 3: Use a native application function
You must produce a report that details the number of registrations for a day and send the report as a PDF to the management team.