

BCS THE CHARTERED INSTITUTE FOR IT

BCS HIGHER EDUCATION QUALIFICATIONS
BCS Level 6 Professional Graduate Diploma in IT

WEB ENGINEERING

March 2015

Answer **any** THREE questions out of FIVE. All questions carry equal marks.
Time: THREE hours.

**Answer any Section A questions you attempt in Answer Book A
Answer any Section B questions you attempt in Answer Book B**

*The marks given in brackets are **indicative** of the weight given to each part of the question.*

Calculators are NOT allowed in this examination.

The figures referenced in this question paper are provided in a separate booklet.

General Comments

The evidence shows that there is a large disparity between well-prepared candidates and those who were not ready to sit this examination.

Future candidates would benefit from taking more care to follow instructions, in particular: when a question asks to illustrate an explanation with examples, these examples usually account for half of the marks allocated to the question.

Section A

Answer Section A questions in Answer Book A

- A1.** You are developing a simple web-based application to manage a collection of web bookmarks.

N.B. This question involves server-side scripting. The preferred language is PHP but answers written in ASP or JSP are also accepted. **Clearly state which server-side scripting language you will be using for the whole question**, and make sure all relevant files are named accordingly.

- a) Web addresses are often referred to as “URL”.

- i) What does this abbreviation mean?

(1 mark)

- ii) A typical URL takes the following format:

```
http://www.example.com/products/index.php?ref=4263&cur=gbp
```

Briefly explain what each of the following parts represent:

- `http://`
- `www.example.com`
- `/products/index.php`
- `?ref=4263&cur=gbp`

(2 marks)

- b) First, you will build a simple front-end in a file named `bookmarkinput.html`

- i) Write HTML to construct the page as indicated in Figure 1.1.

(1 mark)

- ii) Add a simple form to your code. This form must take the following input parameters:

- a single-line string representing the location of a web page, named `address`
- a single-line text description, named `title`

These two parameters must be sent to a script named `savebookmarks.php`. When doing so, the parameters should not be visible in the address bar of the web browser.

(3 marks)

- c) The URL entered in the `address` field might be incorrect.

- i) We wish to add some basic client-side validation to the address field. The validator should check that:

- the first part of the URL is present (`http://` or `https://`)
- the second part contains at least one character

Write the corresponding code.

(3 marks)

- ii) If you were to write code to check the *existence* of a URL, should it be written on the client side, or on the server side? Explain why.

(1 mark)

- d) You have access to a database called `BOOKMARKMANAGER`, which contains a single table named `Bookmarks` (as shown in Figure 1.2). The database is hosted remotely at `mywebhost.com`. The administrator username is `db_admin` and the password is `foobar4263`

Using the server-side scripting language of your choice, write a function called `connectToDatabase`, which:

- takes no input parameters
- establishes an authorised connection with the database host
- gets access to the database itself
- displays relevant error messages when necessary
- returns a connection handle, to be used in future database queries

This function will be saved in a file called `myfunctions.inc`

(3 marks)

- e) You will now write the back-end of your application.

- i) In a file called `savebookmark.php`, write a function called `savebookmark`, which:
- takes 2 strings called `myAddress` and `myTitle` as an input
 - gets access to the `Bookmarks` table described earlier, by referring, via an appropriate mechanism, to the function `connectToDatabase` defined in d)
 - saves `myAddress` and `myTitle` in the `Bookmarks` table
 - returns `true` if the operation was a success, `false` if it failed

(3 marks)

- ii) Still in `savebookmark.php`, write code to:
- generate a full, valid web page entitled "Saving Bookmark"
 - check that the input parameters obtained from the form in `bookmarkinput.html` (see question b) ii)) have been set; if not, display an error message linking back to `bookmarkinput.html`
 - use the `savebookmark` function to save the input parameters in the database
 - display a relevant message regarding the success or failure of the operation

(3 marks)

- f) Write another file called `displaybookmarks.php`:
- the page should display, ordered by `id`, all the web addresses contained in the database
 - the results should appear as a HTML table
 - the first column of the table should contain the title of each bookmark
 - the second column should contain the corresponding address, formatted as a hyperlink

(5 marks)

The following SQL syntax may be useful to accomplish some of this question's tasks:

```
INSERT INTO tbl_name (coll, ...) values (vall, ...);  
SELECT * FROM tbl_name WHERE coll = vall;
```

where `tbl_name`, `coll`, `vall` are replaced with appropriate values.

Answer pointers (A1)

a) i)

Uniform Resource Locator

a) ii)

`http://` is the protocol

www.example.com is the server name

`products/index.php` is the path to the file `index.php`, stored on the server in a folder called `products`

`?ref=4263&cur=gbp` are parameters passed to the script via the GET method

b) i)

```
<html>  
  <head><title>My Bookmark Manager</title></head>  
  <body><h1>Enter new bookmark</h1></body>  
</html>
```

b) ii)

Insert between `</h1>` and `</body>`

```
<form action="savebookmark.php" method="POST">  
  <input type="text" name="address" />  
  <input type="text" name="title" />  
  <input type="submit" />  
</form>
```

c) i)

Using JavaScript and regular expressions:

```
<HTML>
```

```

<HEAD>
<SCRIPT language="JavaScript">
function validateURL() {
    var myURL=document.getElementById("address").value;
    var pattern= new RegExp("https?://.+");
    if (pattern.test(myURL))
    {
        return true;
    }
    else
    {
        window.alert("URL invalid. Try again.");
        return false;
    }
}
</SCRIPT>
</HEAD>
<BODY>
<FORM ... onSubmit="return validateURL();">
...
</FORM>
</BODY>
</HTML>

```

c) ii)

The verification code would have to be written on the server side (e.g., in PHP, using `file_get_contents`).

For security reasons (to avoid cross site scripting attacks), code executed on the client side (e.g. AJAX) cannot query remote servers.

d)

Using PHP5

```

function connectToDatabase()
{
    $connectionHandle =
mysqli_connect("mywebhost.com","db_admin","foobar4263","BOOKMARK
MANAGER");

    if (!$connectionHandle)
    {
        echo "Failed to connect to database: " .
mysqli_connect_error();
    }

    return $connectionHandle;
}

```

e) i)

```

// SaveBookmark.php
include myfunctions.inc

```

```

function saveBookmark($myAddress, $myTitle)
{
    $con = connectToDatabase();
    if (!$con) return false;

    $success = mysqli_query($con, "INSERT INTO BOOKMARKS (address,
title) VALUES ($myAddress, $myTitle, $currentDateTime)");

    mysqli_close($con);
    return $success;
}

```

e) ii)

```

// After e) i)
<HTML>
<HEAD><TITLE>Saving bookmark</TITLE></HEAD>
<BODY>
<?php

if (!isset($_POST["title"]) || !isset($_POST["address"]))
{
    echo "Error : Title must be set and address must not be
empty<br>";
    echo "<a href='bookmarkinput.html'>Back to bookmark input
page</a>";
}
else {
    $success = saveBookmark($_POST["address"], $_POST["title"] )

    if ($success)
    {
        echo "Bookmark saved";
    }
    else
    {
        echo "Failed to save bookmark";
    }
}
?>
</BODY>
</HTML>

```

f)

```

// displaybookmarks.php
include myfunctions.inc

function displayBookmarks()
{
    $con = connectToDatabase();
    if (!$con)
    {

```

```

        echo "Could not connect to database";
        return;
    }

    $result = mysqli_query($con, "SELECT address, title from
bookmarks ordered by id");

    if ($result)
    {
        echo "Failed to query database";
        mysqli_close($con);
        return;
    }

    if (!mysqli_num_rows($result))
    {
        echo "No bookmarks to display";
        mysqli_close($con);
        return;
    }

    echo "<table><tr><th>Title</th><th>address</th></tr>";

    while($row = mysqli_fetch_array($result))
    {
        echo "<tr>";
        echo "<td>" . $row['Title'] . "</td>";
        $address = $row['Address'];
        echo "<td><a href=\"\$address\">$address</a></td>";
        echo "</tr>";
    }
    echo "</table>";

    mysqli_close($con);
}

<HTML>
<HEAD><TITLE>Displaying bookmarks</TITLE></HEAD>
<BODY>
<?php
    displayBookmarks();
?>
</BODY>
</HTML>

```

Examiners' Guidance Notes (A1)

a) i)

The evidence suggests that this question was not well answered. A small percentage of answers were entirely accurate or contained only minor errors (e.g. spelling mistakes). A large percentage of answers contained numerous errors. Typical errors involved:

- Unique or Universal (instead of Uniform)
- Reference or Relation (instead of Resource)
- Link or Language (instead of Locator)

a) ii)

This question received mixed answers. The first two parts were answered well, although some answers were sometimes vague.

The third part was not answered as well, and the evidence suggests that answers were too vague (e.g. “page location”) and failed to express that the string represents a path to a file, stored on the server within a specific directory.

The fourth part was rarely understood (but when it was understood, was also usually well explained). Common misunderstandings involved:

- referring to the string as some form of encryption or encoding
- mistaking the parameters for a login and password

b) i)

The question was generally well answered. Most answers were either entirely correct, or contained only a few errors, such as:

- missing closing tags
- poorly formed document (e.g. body tag inside of the head tag)
- confusion around the role and location of the title tag
-

A small number of answers received no marks, as they contained errors (including poorly formed and/or incorrectly named tags, which demonstrated a lack of understanding of basic HTML syntax)

b) ii)

The question was mostly well answered, with some errors.

Common errors included:

- missing submit button
- missing form closing tag
- incorrect value for the name attribute
- incorrect input type (e.g. textarea, with incorrect syntax)

c) i)

The question was generally poorly answered (although a few candidates received full marks)

Common errors or omissions included:

- answer written in PHP (not a client side language)
- omitting the mechanism by which a javascript method is called from an HTML form
- approximative javascript syntax (especially function declaration)
- approximative regular expression syntax
- no user feedback (e.g. alert message) and no return statement

Although the answer pointer uses javascript and regular expressions, alternative methods were acceptable, e.g.:

- using HTML5 to validate the URL
- using javascript, but not regular expressions (the structure of the URL could be analysed by splitting it into various substrings, and comparing their value or length to the expected pattern)

c) ii)

The question was generally poorly answered (although some candidates received full marks).

Many answers assumed the verification had to be done on the client side, or on both sides, and received no marks.

When the first part of the answer was correct, candidates usually could not explain convincingly why the verification had to be performed on the server side. Most explanations focused on generic properties of server-side validation (which were irrelevant in the context on the question) and did not mention the specific limitations of client-side validation, which (in this particular case) made server side validation the only possible option.

d)

The question was relatively well answered. Most candidates could write code to connect to the database. However, answers were often incomplete concerning:

- function declaration
- return value
- database error handling

A small number of answers were off topic, as they contained database queries and performed tasks which were not relevant to the question.

e) i)

The question was generally poorly answered (although a few candidates received full marks). The query itself was usually relatively well written.

However, the evidence suggests that many candidates struggled with basic function syntax and usage (handling inputs and outputs and invoking a function).

e) ii)

The question received very mixed answers. Some candidates received full marks, whereas some candidates performed very poorly.

Many candidates struggled with the following tasks:

- interweaving HTML code with PHP code
- writing a HTML link (generally omitted, or badly formed)
- making use of a previously defined function

f)

The question was often answered poorly, although some candidates received full marks.

Common mistakes or omissions involved:

- the overall syntax and mechanism of a PHP loop
- the overall syntax of an HTML table
- the mechanism to access each result row after a successful select query

A2. In February 2014, the online auction house eBay® was the victim of an attack that compromised the details of an estimated 145 million users. Please read the press article on the attack in Figure 2.1 then answer the questions below:

- a)** One possible security risk for the users of eBay® is phishing. A partial solution to this is to never follow hyperlinks to the eBay® website from an email, but instead enter the URL manually.
- i) Explain, with an example, what is meant by the term *phishing*.
(2 marks)
- ii) Aside from phishing, and the attack outlined in the article, give **THREE** other security risks to the users of eBay® that could result in disclosure of user data, and for each risk outline a method to prevent it.
(6 marks)
- b)** It is believed that no financial or credit card data was revealed, as this information is stored separately. With reference to the information in the article, and using a suitable example to demonstrate the impact, explain how the information disclosed might still present a serious risk to affected users.
(3 marks)
- c)** When the attack was discovered, eBay® asked users to change their passwords.
- i) With a suitable example, describe **ONE** positive consequence of this action, and **ONE** negative consequence of this action. (N.B. this may be considered from either the corporate or user perspective).
(2 marks)
- ii) One alternative action that eBay® could have applied would be to disable all user accounts and ask the users to reactivate them after proving their identity.
- Explain why each of the following normal methods of account verification might not be safe:
- asking the user to confirm their current address and payment details
 - sending an activation code to the user's email address
 - confirming the maiden name of the user's mother
- (3 marks)**
- iii) Taking into account the weaknesses of the methods discussed in part c) ii) propose (with appropriate justification) a robust method of verifying the identity of a user that offers a good balance between user account safety and cost effectiveness.

(3 marks)

- d) Initial communications from eBay® about the security compromise was made in the form of a press release on their corporate website, two weeks after discovery.

In no more than 500 words, discuss eBay's® actions.

In your answer, you should consider the pros and cons of both the method of communication used and the time interval between discovery and initial communication.

(6 marks)

Answer pointers (A2)

a) i)

Phishing is attempting to obtain account credentials through fake emails or websites, but which seem superficially to come from legitimate sources [1 mark]. Candidates should provide a credible real-life example [1 mark].

a) ii)

Sample risks (with possible mitigation in brackets) include:

- Brute forcing of passwords (locking out accounts after a number of unsuccessful attempts)
- Packet sniffing to reveal plaintext passwords (use encryption)
- Keylogger on client machine to capture username/password (check physical hardware and/or run a rootkit checker/antivirus)
- SQL injection attack (ensure all user inputs are validated on the server side) to gain elevated access on the server and access data directly
- Spyware on client machine (use antivirus/spyware checker)
- Not destroying letters with sensitive information e.g. passwords (shred confidential documents when disposing)

[1 mark for each *distinct* risk to data, 1 mark for suitable solution]

b)

The key risk here is identity theft, which is using sensitive information to fraudulently acquire ID documents (e.g. passport) in the victim's name [1 mark for definition and explanation], which can then be used to acquire bank accounts, loans, credit cards, etc. [1 mark]. Candidates should indicate the relevant sensitive data *that was listed in the article* that was compromised that could be used for this (customer's name, email address, physical address, phone number and date of birth), and mention that passwords and financial details is not required to attempt this attack. [1 mark]

c) i)

Candidates should list one positive [1 mark] and one negative [1 mark] consequence with a good example, consequences listed must be relevant to the question. Some sample consequences might include:

Positive	Negative
No inconvenience to users by forcibly locking out their accounts and therefore denying them access to the service until they can unlock the account.	Users do not have to change their passwords, so brute forcing the encrypted passwords over time may still yield account details that can be used on eBay itself.
Users feel empowered by choosing when (and if) to change their password, based on their own personal assessment of the risk to their data.	The request to change passwords does not mitigate against other risks, e.g. possible identity theft, or that the password might be used on other systems (e.g. the email provider).

c) ii)

Answers will be marked on merit, with 1 mark for each section given for a well-reasoned justification. An example of possible answers:

- Asking the user to confirm their current address and payment details.

The address details are already compromised, which means that an attacker only needs indication of financial details to be able to steal the user's account. This could be achieved by e.g. dumpster diving for credit card numbers (made even easier since they have the address of the card holder).

- Sending an activation code to the user's email address.

Many users have the same password on multiple systems, so there are likely to be users who re-use the password from their email account on eBay. In that instance, if the attackers can break the eBay password (e.g. by using rainbow tables) they would be able to gain access to the user's email account as well, and so could intercept the activation code.

- Confirming the user's mother's maiden name.

Unless eBay already have the mother's maiden name on record (and it was not listed as part of the compromised data), it is not feasible for eBay to know the answer to this question. Furthermore, in the UK, knowing a user's name, date of birth and address may allow them to find out the user's mother's maiden name from the register of births (if the user lives in the area they were born).

c) iii)

An open question, 1 mark for the outline of the solution, 2 marks for justification and fitness for purpose of the solution. Candidates must propose a method that has not been mentioned elsewhere in the question, and that either does *not* suffer from the weaknesses in part c) ii) or is acknowledged and with a justification as to why this is not important.

Answers that focus too much on either side of system security (with impacts on either cost or usability), or aiming for a cheap solution (with impacts on security) will be capped at a maximum of 1 mark, unless they provide a very good justification as

to why the decision should be taken, and include some acknowledgment of the weaknesses.

d)

This is a discussion question. Marks will be allocated for:

- the depth of discussion on each of the two aspects to be considered (method used, and time taken from discovery to disclosure) [2 marks for each aspect], with full marks granted on each aspect for a balanced consideration (including both positive and negative views).
- ii) the structure and completeness of the discussion overall [1 mark]
- iii) stating a definite answer to the question [1 mark]; this is marked on the quality of this conclusion and degree to which it is supported by the main argument.

Examiners' Guidance Notes (A2)

a) i)

This question was generally well answered.

Occasionally answers would not provide a suitable example, but instead give an abstract description or simple rephrasing of the definition.

a) ii)

This question was generally well answered.

However, a number of answers would refer to risks that would not result in disclosure of sensitive data, e.g. spam, adware. Some answers provided risks that were not sufficiently distinct – e.g. providing virus, keylogger and spyware (where the virus element is only relevant when it is the delivery mechanism for a keylogger or for spyware, both of which are about collection of user entered data and are therefore too similar – this would only count as a single risk). Finally, some answers provided more than the required number of risks, which were not assessed.

b)

This question was generally poorly answered, though a few candidates achieved full marks.

Whilst answers generally indicated some basic risks related to the disclosed data (e.g. using email addresses to sell to spammers), very few candidates correctly identified the most serious risk of identity theft.

c) i)

The question was very poorly answered, with the vast majority of answers not considering the optional nature of eBay asking users to change passwords, and instead focussing on the consequences of the attack in general – since these consequences occur regardless of eBay's choice whether to ask users or instead to more forcibly lock accounts (or take other action), these did not get any credit.

c) ii)

The question received mixed answers. Generally the analysis of the first option was well answered, but the other two were much weaker, and did not adequately assess how attackers would get the information required for a viable attack – e.g. how would attackers access the users email when the password disclosed was a) for eBay and b) encrypted.

c) iii)

The question was generally poorly answered, or entirely omitted.

Very often answers provided a solution that was heavy on security and did not consider the practicality of the proposed solution and did not attempt to justify or explain the high costs involved nor the arduous processes. References were often made to two-factor authentication with no indication of how this would solve the problem.

d)

The question was generally poorly answered.

Many answers focussed only on the time taken by eBay to respond, and generally only focused on the negative aspects of their actions. Finally, there was often no conclusion provided.

Section B

Answer Section B questions in Answer Book B

- B3. a)** “A well formed XML document is necessarily valid”. Briefly explain the correctness or otherwise of this statement. Support your answer with a suitable example. **(4 marks)**
- b)** Data interchange formats between applications can be based on XML or JSON technology. Discuss the appropriateness of each, citing suitable examples. **(4 marks)**
- c)** XML documents can be validated against DTDs or XML schema. Discuss the benefits and drawbacks of each approach citing suitable examples. **(4 marks)**
- d)** Write an external DTD, `company.dtd` for the following XML file:
- Sequence of elements is as shown in the XML code in fig 3.1 (i.e. CompanyList is a container of Company elements)
 - The attribute Company Code is mandatory
 - Company_url is optional
 - Company_url must have a page attribute
 - For each Company, one Company_name, Company_url, Contact_name and zero or more Job_vacancy element must be present
 - For each Contact, there must be a Contact_name, Contact_phone, Contact_email
 - For each Job_vacancy, exactly one Job_title, Job_description, Job_location, Job_salary and Start_date must be present
- (13 marks)**

Answer pointers (B3)

a)

The statement is false and whilst it can be well formed it is not true that it is valid. A valid XML document is necessarily well formed.

b)

XML vs JSON:

- XML is easily readable by both humans and machines
- JSON is easier to read for both humans and machines.
- XML is object-oriented
- Actually, XML is document-oriented. JSON is data-oriented. JSON can be mapped more easily to object-oriented systems.
- XML is being widely adopted by the computer industry
- JSON is just beginning to become known. Its simplicity and the ease of converting XML to JSON makes JSON ultimately more adoptable.

c)

DTD vs XML Schema:

- DTD's are not namespace aware.
- DTD's have #define, #include, and #ifdef – or, less C-oriented, the ability to define shorthand abbreviations, external content, and some conditional parsing.
- A DTD describes the entire XML document (even if it leaves "holes"); a schema can define portions.
- XSD has a type system.
- XSD has a much richer language for describing what element or attribute content "looks like." This is related to the type system.
- You can put a DTD inline into an XML document; you cannot do this with XSD.
- You have far more control over what is considered a valid XML document using a schema.
- You can even extend your types from other types you've created, require uniqueness within scope, and provide lookups.

d)

```
1 <?xml version = "1.0" encoding="ISO-8859-1"?>
2 <!ELEMENT CompanyList (Company*)>
3 <!ELEMENT Company
  (Company_name,Company_url,Contact,Job_vaccancy*)>
4 <!ATTLIST Company CODE CDATA #REQUIRED>
5 <!ELEMENT Company_name (#PCDATA)>
6 <!ELEMENT Company_url EMPTY>
7 <!ATTLIST Company_url page CDATA #REQUIRED>
8 <!ELEMENT Contact
  (Contact_name,Contact_phone,Contact_email)>
9 <!ELEMENT Contact_name (#PCDATA)>
10 <!ELEMENT Contact_phone (#PCDATA)>
11 <!ELEMENT Contact_email (#PCDATA)>
12 <!ELEMENT Job_vacancy (Job_title,
  Job_description,Job_location, Job_salary, Start_date)>
13 <!ELEMENT Job_title (#PCDATA)>
14 <!ELEMENT Job_description (#PCDATA)>
15 <!ELEMENT Job_location (#PCDATA)>
16 <!ELEMENT Job_salary (#PCDATA)>
17 <!ELEMENT Start_date (#PCDATA)>
```

Examiners' Guidance Notes (B3)

This question was very popular with candidates.

Most candidates knew the difference between a well formed and a valid xml document and managed to answer the question with suitable examples.

Few candidates were able to articulate the benefits and drawbacks of using XML and JSON for data interchange. Examples cited were weak.

Many candidates were able to clearly explain the differences in validation using DTDs and XML schemas with good examples.

Most candidates scored well on writing DTD for part (d). Some candidates did not correctly handle the mandatory attributes nor the optional elements.

Overall, majority of the candidates were well prepared for this question.

- B4. a)** An independent blogger has decided to publish properties for sale on her site based on data accumulated from various estate agents. She has checked that there are no copyright breaches, as she is providing a free service. She has created an XML document to hold the details of these properties; a sample is shown in fig 4.1. She wants to display these properties as shown in figure 4.2.

The properties are separated into houses and flats with details of location, image, agent, local authority and the condition of the property. The image is to also serve as a link to a photo gallery for the property. The agent name is a link to the agent's web site and the email address is to be an active link. The style sheet has already been developed including an HTML template, provided in Figure 4.3.

You are to write the code that will process the XML data and render it as a web page as shown in Figure 4.3. You are advised to show the code for the following, including appropriate comments to explain how the code will work:

- i.* The code to display all the properties by type;
- ii.* The code to display the property image;
- iii.* The code to make the displayed image into a link that will open another window where a photo gallery of the property will be displayed;
- iv.* The code to display all the agent details as shown including the agent name and the email address rendered as links.

(18 marks)

- b)** A further enhancement that the blogger would like to provide is a feature for a potential buyer to make an appointment with the agent to view the property. Discuss how this could be achieved; note that you are not required to code this.

(4 marks)

- c)** List two further enhancements that could be made to this site to make it more user friendly and fit for purpose.

(3 marks)

Answer pointers (B4)

a)

```
<table border="1">
  <xsl:for-each select="propertyList/category">
    <tr>
      <th colspan="6" align="center">
        <div class="alttd">
          <xsl:value-of select="type"/>
        </div>
      </th>
    </tr>
    <tr bgcolor="pink">
      <th align="center">Reference</th>
      <th align="center">Location</th>
      <th align="center">Image</th>
      <th align="center">Agent</th>
      <th align="center">Local Authority</th>
      <th align="left">Condition</th>
    </tr>
    <xsl:for-each select="property">
      <tr>
        <td>
          <b>
            <xsl:value-of select="@id"/>
          </b>
        </td>
        <td>
          <xsl:value-of select="location"/>
        </td>
        <td>
          <a href="{property_gallery/@page}"
target="_blank">
            <img>
              <xsl:attribute name="src">
                <xsl:value-of select="image/@src" />
              </xsl:attribute>
              <xsl:attribute name="width">
                <xsl:value-of
select="image/@width" />
              </xsl:attribute>
              <xsl:attribute name="height">
                <xsl:value-of
select="image/@height" />
              </xsl:attribute>
            </img>
          </a>
        </td>
        <td>
          <xsl:for-each select="agent">
            <a href="{agent_url/@page}"
target="_blank">
              <xsl:value-of select="name"/>
            </a>
            <br/>
            Address:<xsl:value-of select="address"/>
            <br />
            email:<a href="mailto:{email}">
              <xsl:value-of select="email"/>
            </a>
            <br />
            Phone: <xsl:value-of select="phone"/>
          </xsl:for-each>
        </td>
        <td>
          <xsl:value-of select="local_authority"/>
        </td>
        <td>
          Year of build:<xsl:value-of
select="@year_built"/>
          <br />
          <xsl:value-of select="condition"/>
        </td>
      </tr>
    </xsl:for-each>
  </xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

b)

Make the email link launch a calendar where the user can select the time and send the email with details, etc

c)

Website could provide additional information regarding the area in terms of crime rates, etc. Price information and links to other sites that provide details of the properties sold in that area. Site to be accessible from any device and browser.

Examiners' Guidance Notes (B4)

This question tested candidate's knowledge to render an xml document as an html page.

Candidates were required to use *XSLT for loops*; an outer loop to process the property category and inner for loop to process through each property in that category. There was further need to code so that the images were correctly rendered and would act as a link to a gallery of additional images for that property.

There was also a need to render agent details correctly, url and email.

Overall, there were some excellent solutions, however, many found the coding a challenge and some just wrote html code.

Parts (b) and (c) were generally well done with some very good suggestions on improving the usability of the site.

- B5. a)** Define and briefly describe a Web feed using a suitable example. **(2 marks)**
- b)** Define and briefly describe a Web API using a suitable example. **(3 marks)**
- c)** A Web API can be implemented using either a Representational State Transfer (REST) protocol or Simple Object Access Protocol (SOAP). Describe the workings of each of these protocols and benefits of each. **(5 marks)**
- d)** Identify the stages and issues when creating an API from a producer's viewpoint. **(5 marks)**
- e)** As a Web engineer, identify at least FIVE major challenges in the design and development of an image-hosting site, where users can upload their images to a central server and the images can be retrieved via a web link or an API. Initially, focus on two the ability to upload (write) an image to the server, and the ability to query for an image. For each of the challenges identified, provide an outline solution including the technology to be used. Your answer should be in the form of a slide presentation to the client (5 slides with bullet points). **(10 marks)**

Answer pointers (B5)

a)

Small modules of information that can be plugged into existing websites, consumed by clients on their devices or by aggregators to be presented by users with other feeds. Feeds relatively static and it is the same information for all subscribers.

b)

APIs can be large programs that provide customized, detailed information. They are akin to a class in OO, response is based on the information provided in the request but without any further exchange to clarify any of the supplied information, i.e. a result is returned.

c)

Discussion involving comparison with GET and POST and transmission of requests that are completely visible or encoded, etc

d)

Planning

Enabling authentication and encryption

Deciding between REST and SOAP

Performance considerations

Error handling

e)

Availability; Performance; Reliability; Scalability; Manageability; Cost

Examiners' Guidance Notes (B5)

This was the least popular question.

Most candidates were able to answer correctly part (a).

Most candidates struggled to answer parts (b) and (c), not recognising that it required a discussion of different technologies and how best to meet the challenges of designing an API.

Some candidates used a model of SDLC for part(d) which does not fully address the stages that are necessary in the development of an API.

There were some good solutions for part(e) which covered the areas of performance, security, cost, etc.