BCS THE CHARTERED INSTITUTE FOR IT

BCS HIGHER EDUCATION QUALIFICATIONS
BCS Level 5 Diploma in IT

WEB APPLICATION DEVELOPMENT

SAMPLE PAPER AND ANSWER POINTERS

Answer **any** FOUR questions out of SIX. All questions carry equal marks.
Time: TWO hours

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

Calculators are NOT allowed in this examination.

Section A
Answer Section A questions in Answer Book A

Section B
Answer Section B questions in Answer Book B

Section A

Answer Section A questions in Answer Book A

A1.

a) Compare and contrast XML and JSON.

(10 marks)

b) Explain why JSON gained popularity over XML in web application development?

(5 marks)

c) Describe the MVC architecture.

(10 marks)

A1 Answer Pointers

a) [Syllabus coverage 1.1]

XML is a markup language and JSON is a data interchange language so they are not directly comparable. XML can however be used for data interchange and its use for this function may be compared with JSON.

(2 marks)

Comparison of JSON and XML

Structure

XML represents data as a tree structure whereas JSON uses a map with key value pairs.

Clarity

JSON is generally considered to be easier for hums to read and understand.

File size

In general, the same date represented in JSON will occupy less space than the XML representation.

Processing Seed

In general, JSON can be parsed quicker than XML with less computational resource usage.

Namespace Support

Namespaces can be used in XML to distinguish between properties that are different but have the same name. JSON does not have namespace support.

Array Support

JSON implements arrays but XML only implements them indirectly.

Complex types

XML supports many complex data types whereas JSON is limited to basic types.

Security

Both JSON and XML can be implemented in ways which make them insecure.

2 marks for each criterion cited when accompanied by a brief description. The list above is not exhaustive and candidates will receive credit for citing any other relevant factors.

b) [Syllabus coverage 1.1]

The vast majority of front end development of web applications makes use of Javascript. Since JSON is directly understandable to a Javascipt program this means it is often preferred in contrast to XML which must be parsed before the data it is carrying can be used. As the number of back frameworks which use Javascript have grown this has increasingly led to the use of JSON on the server. In general, JSON is preferred wherever the additional features implemented by XML are deemed to be unnecessary.

(5 marks)

c) [Syllabus coverage 1.4]
From https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm

*The **Model-View-Controller (MVC)** is an architectural pattern that separates an application into three main logical components: the **model**, the view, and the controller. Each of these components are built to handle specific development aspects of an application. MVC is one of the most frequently used industry-standard web development framework to create scalable and extensible projects.*

Model
The Model component corresponds to all the data-related logic that the user works with. This can represent either the data that is being transferred between the View and Controller components or any other business logic-related data. For example, a Customer object will retrieve the customer information from the database, manipulate it and update it data back to the database or use it to render data.

View
The View component is used for all the UI logic of the application. For example, the Customer view will include all the UI components such as text boxes, dropdowns, etc. that the final user interacts with.

Controller
Controllers act as an interface between Model and View components to process all the business logic and incoming requests, manipulate data using the Model component and interact with the Views to render the final output. For example, the Customer controller will handle all the interactions and inputs from the Customer View and update the database using the Customer Model. The same controller will be used to view the Customer data.

(10 marks)

A2.

a) Give descriptions of the following two software development methods:
   i. Kanban;
   ii. Scrum.

(20 marks)

b) List **FIVE** potential advantages of using Scrum and/or Kanban as part of a web application development process.

(5 marks)

A2 Answer Pointers

a) [Syllabus coverage 2.3]

i)

Kanban grew out of the lean manufacturing movement and was a technique pioneered by Toyota. The basic idea is that smoothing the flow of work can make production more efficient. Kanban is a general set of principles in contrast to the more rigidly defined practice in Scrum. The basic principles are:

- *Make all work visible;*
- *Limit the amount of work in progress;*
- *workflow Practice continual self improvement.*

The work of all kanban teams revolves around a kanban board, a tool used to visualize work and optimize the flow of the work among the team. While physical boards are popular among some teams, virtual boards are a crucial feature in any agile software development tool for their traceability, easier collaboration, and accessibility from multiple locations.

Regardless of whether a team's board is physical or digital, their function is to ensure the team's work is visualized, their workflow is standardized, and all blockers and dependencies are immediately identified and resolved. A basic kanban board has a three-step workflow: To Do, In Progress, and Done. However, depending on a team's size, structure, and objectives, it can be mapped to meet the unique process of any particular team.

The main purpose of representing work as a card on the kanban board is to allow team members to track the progress of work through its workflow in a highly visual manner. Kanban cards feature critical information about that particular work item, giving the entire team full visibility into who is responsible for that item of work, a brief description of the job being done, how long that piece of work is estimated to take, and so on. Cards on virtual kanban boards will often also feature screenshots and other technical details that is valuable to the assignee. Allowing team members to see the state of every work item at any given point in time, as well as all of the associated details, ensures increased focus, full traceability, and fast identification of blockers and dependencies.

ii)

From https://www.mountaingoatsoftware.com/agile/scrum:

The Scrum model suggests that projects progress via a series of sprints. In keeping with an agile methodology, sprints are timeboxed to no more than a month long, most commonly two weeks.

Scrum methodology advocates for a planning meeting at the start of the sprint, where team members figure out how many items they can commit to, and then create a sprint backlog – a list of the tasks to perform during the sprint.

During an agile Scrum sprint, the Scrum team takes a small set of features from idea to coded and tested functionality. At the end, these features are done, meaning coded, tested and integrated into the evolving product or system.

On each day of the sprint, all team members should attend a daily Scrum meeting, including the ScrumMaster and the product owner. This meeting is timeboxed to no more than 15 minutes. During that time, team members share what they worked on the prior day, will work on that day, and identify any impediments to progress.

The Scrum model sees daily scrums as a way to synchronize the work of team members as they discuss the work of the sprint.

At the end of a sprint, the team conducts a sprint review during which the team demonstrates the new functionality to the PO or any other stakeholder who wishes to provide feedback that could influence the next sprint.

This feedback loop within Scrum software development may result in changes to the freshly delivered functionality, but it may just as likely result in revising or adding items to the product backlog.

Another activity in Scrum project management is the sprint retrospective at the end of each sprint. The whole team participates in this meeting, including the ScrumMaster and PO. The meeting is an opportunity to reflect on the sprint that has ended, and identify opportunities to improve.

(20 marks)

b) [Syllabus coverage 2.3 – 2.4]

- *Transparency*
- *Early and predictable delivery*
- *Predictable costs*
- *Opportunities for change*
- *Focus on business value*
- *Focus on users*
- *Improved quality*

(5 marks)

A3.

a) Discuss the potential uses for JavaScript in front end web application development.

(5 marks)

b) Describe the relationship between jQuery and JavaScript

(5 marks)

c) Discuss the importance of a front end framework with which you are familiar in developing web applications

(15 marks)

A3 Answer Pointers

a) [Syllabus coverage 3.1]

From Wikipedia:

JavaScript is an interpreted programming language that conforms to the ECMAScript specification. JavaScript is high-level, often just-in-time compiled, and multi-paradigm. It has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.

Alongside HTML and CSS, JavaScript is one of the core technologies of the World Wide Web.] JavaScript enables interactive web pages and is an essential part of web applications. The vast majority of websites use it for client-side page behaviour, and all major web browsers have a dedicated JavaScript engine to execute it.

As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative programming styles. It has application programming interfaces (APIs) for working with text, dates, regular expressions, standard data structures, and the Document Object Model (DOM). However, the language itself does not include any input/output (I/O), such as networking, storage, or graphics facilities, as the host environment (usually a web browser) provides those APIs.

(5 marks)

b) [Syllabus coverage 3.2]

From https://www.c-sharpcorner.com/article/javascript-vs-jquery-difference-between-javascript-and-jquery/

JavaScript	***jQuery***
A weakly typed dynamic programming language	*A fast and concise JavaScript Library*
A scripting language for interface interactions and for controlling the document content	*A framework which makes event handling, animating, and Ajax interactions simpler and faster*
An interpreted language	*Uses resources given by JavaScript to make things easier*

Need to write your own scripting which may take time	*need not write much scripting which already exists in jQuery*
Developers need to handle multi-browser compatibility by writing their own JavaScript code	*Is a multi-browser JavaScript library which reduces the work of developers during deployment*
Developers are prone to make many common browser-related errors.	*Developers don't have to worry about browser compatibility issues.*
Need not include anything to work in browser as all modern browsers support JS	*Need to include the jQuery library URL in the header of the page*
More lines of code	*Fewer lines of code*
Faster in accessing DOM	*Suited for complex operations where developers are prone to mistakes and writing poor lines of code.*

Advantages of jQuery

- *You can code most common JS actions using jQuery with fewer lines of code.*
- *Browser compatibility – you can write code which runs across browsers without having to know the various browser intricacies and won't break.*
- *Lets you write JavaScript quicker and easier.*
- *Avoids common browser errors*
- *Simplification of usually complicated operations – complex operations like Ajax interactions, animation, event handling etc. are handled by jQ with the best lines of code.*
- *jQ is battle tested and uses the fast and best lines of code for accomplishing most tasks.*

(5 marks)

c) [Syllabus coverage 3.4]

Advantages of front end frameworks include:

1. ***Maintainability**: Breaking up your app into reusable, standalone components makes it easier to make quick changes that don't impact the rest of the application.*
2. ***Separation of concerns:** Modern framework design encourages a maintainable, modular architecture and allows your front-end developers to focus on what they do best: taking data and displaying it to users in an intuitive and efficient way.*
3. ***Speed:** Boilerplate code aimed at addressing common problems makes it easier for you to get your app up and running; component-based design makes it quicker to develop.*
4. ***Collaboration:** Since frameworks often follow similar design patterns, it's easier for developers who are new to your codebase to develop and maintain your app.*
5. ***Community:** Popular frameworks have a community of people around them with dedicated tutorials, forums, meetups, and generally supportive developers you can seek help from.*

From Wikipedia:

The AngularJS framework works by first reading the Hypertext Markup Language (HTML) page, which has an additional custom HTML attributes embedded into it. Angular interprets those attributes as directives to bind input or output parts of the page to a model that is represented by standard JavaScript variables. The values of those JavaScript variables can be manually set within the code, or retrieved from static or dynamic JSON resources.

AngularJS is built on the belief that declarative programming should be used to create user interfaces and connect software components, while imperative programming is better suited to defining an application's business logic. The framework adapts and extends traditional HTML to present dynamic content through two-way data-binding that allows for the automatic synchronization of models and views. As a result, AngularJS de-emphasizes explicit Document Object Model (DOM) manipulation with the goal of improving testability and performance.

AngularJS's design goals include:

- *to decouple DOM manipulation from application logic. The difficulty of this is dramatically affected by the way the code is structured.*
- *to decouple the client side of an application from the server-side. This allows development work to progress in parallel and allows for reuse of both sides.*
- *to provide structure for the journey of building an application: from designing the UI, through writing the business logic, to testing.*

AngularJS implements the MVC pattern to separate presentation, data, and logic components. Using dependency injection, Angular brings traditionally server-side services, such as view-dependent controllers, to client-side web applications. Consequently, much of the burden on the server can be reduced.

(15 marks)

Section B

Answer Section B questions in Answer Book B

B4.

a) Describe at least **FOUR** of the main goals and benefits of a service-oriented architecture.

(10 marks)

b) How might a service oriented architecture concept be implemented using REST?

(15 marks)

B4 Answer Pointers

a) [Syllabus coverage 4.2]

***Basic**: mentioning a subset (4 of) of the following goals and benefits: Increased Federation, Increased Intrinsic Interoperability, Increased Vendor Diversity Options, Increased Business and Technology alignment, Increased ROI, Reduced IT Burden, Increased Organizational Agility*

***Stronger:** Mentioning 6 or more of the goals and benefits*
***Exemplary**: mentioning 4 or more of the goals and benefits and their definition. For the definitions and how SOA realizes them please refer to:*

Thomas Erl. "Service-Oriented Architecture: Analysis and Design for Services and Microservices (The Prentice Hall Service Technology Series from Thomas Erl)". Chapter 3.4 Goals and benefits of Service Oriented Architecture

(10 marks)

b) [Syllabus coverage 4.3]

***Basic**: REST (Representational state transfer (REST) is a software architectural style that defines a set of constraints to be used for creating Web services. Web services that conform to the REST architectural style, called RESTful Web services, provide interoperability between computer systems on the Internet. RESTful Web services allow the requesting systems to access and manipulate textual representations of Web resources by using a uniform and predefined set of stateless operations.)*
***Stronger**: Mentioning REST architectural properties:*
Performance, Scalability, simplicity of a uniform interface, Technology support, portability of components (loos coupling), reliability in the resistance to failure at the system level
***Exemplary**: explaining the rest architectural properties in more detail:*
For a complete discussion about these architectural properties please refer to Jim Webber. "REST in Practice"- section "Web as an application platform"

(15 marks)

B5.

a) Explain the benefits of Test Driven Development (TDD)

(10 marks)

b) Discuss the use of web analytics in the evaluation of a web-based application

(15 marks)

B5 ANSWER POINTERS

a) [Syllabus coverage 5.1 - 5.2]

***Basic**: Definition of test driven development (i.e. Test-driven development (TDD) is a software development process that relies on the repetition of a very short development cycle: requirements are turned into very specific test cases, then the code is improved so that the tests pass. This is opposed to software development that allows code to be added that is not proven to meet requirements.)*
Stronger**: Explaining the TDD lifecycle (i.e. 1. **Add a test
In test-driven development, each new feature begins with writing a test. Write a test that defines a function or improvements of a function, which should be very succinct. To write a test, the developer must clearly understand the feature's specification and requirements. The developer can accomplish this through use cases and user stories to cover the requirements and exception conditions, and can write the test in whatever testing framework is appropriate to the software environment. It could be a modified version of an existing test. This is a differentiating feature of test-driven development versus writing unit tests after the code is written: it makes the developer focus on the requirements before writing the code, a subtle but important difference.
*2. **Run all tests and see if the new test fails***
This validates that the test harness is working correctly, shows that the new test does not pass without requiring new code because the required behavior already exists, and it rules out the possibility that the new test is flawed and will always pass. The new test should fail for the expected reason. This step increases the developer's confidence in the new test.
*3. **Write the code***
The next step is to write some code that causes the test to pass. The new code written at this stage is not perfect and may, for example, pass the test in an inelegant way. That is acceptable because it will be improved and honed in Step 5.
At this point, the only purpose of the written code is to pass the test. The programmer must not write code that is beyond the functionality that the test checks.
*4. **Run tests***
If all test cases now pass, the programmer can be confident that the new code meets the test requirements, and does not break or degrade any existing features. If they do not, the new code must be adjusted until they do.
*5. **Refactor code**)*
***Exemplary**: Explaining TDD and the lifecycle with a simple example (for instance by assuming that they have been told to implement a simple feature (for instance we are asked to write a function that receives to integers and sums them up F(x, y) = x + y, we first write a test that gets for instance 4 and 6 and must return 10, we*

now run the test, as we have not written the function it must fail, now we write the function (in language of choice for instance int f (int x, int y) {return x+y};) we now run the test again. The test must pass now, if it does not we must refactor the code until it does.

(10 marks)

b) [Syllabus coverage 5.4]

***Basic**: a short explanation of web analytics is and how they can be used (i.e. Web analytics is the measurement, collection, analysis and reporting of web data for purposes of understanding and optimizing web usage.*

***Stronger**: more through discussion (i.e. However, Web analytics is not just a process for measuring web traffic but can be used as a tool for business and market research, and to assess and improve the effectiveness of a website. Web analytics applications can also help companies measure the results of traditional print or broadcast advertising campaigns. It helps one to estimate how traffic to a website changes after the launch of a new advertising campaign. Web analytics provides information about the number of visitors to a website and the number of page views. It helps gauge traffic and popularity trends which is useful for market research.)*

***Exemplary**: providing a number of examples of how web analytics can help evaluating and improving a web-based application (i.e. for instance we can use web analytics to find out the popularity of a specific product that we have recently launched in our website. We can measure the number of views, demography of the visitors (which countries, what time of the day…)*

(15 marks)

B6.

a) List **FIVE** of the OWASP top ten security issues.

(5 marks)

b) Choose **ONE** of the issues you gave as an answer to part a) and give a detailed description of it.

(5 marks)

c) Describe threat modeling and explain why it is one of the most important steps in developing secure web applications.

(15 marks)

B6 Answer Pointers

a) [Syllabus coverage 6.1 – 6.2]

1. ***Injection.*** *Injection flaws, such as SQL, NoSQL, OS, and LDAP injection, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.*
2. ***Broken Authentication****. Application functions related to authentication and session management are often implemented incorrectly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users' identities temporarily or permanently.*
3. ***Sensitive Data Exposure****. Many web applications and APIs do not properly protect sensitive data, such as financial, healthcare, and PII. Attackers may steal or modify such weakly protected data to conduct credit card fraud, identity theft, or other crimes. Sensitive data may be compromised without extra protection, such as encryption at rest or in transit, and requires special precautions when exchanged with the browser.*
4. ***XML External Entities (XXE)****. Many older or poorly configured XML processors evaluate external entity references within XML documents. External entities can be used to disclose internal files using the file URI handler, internal file shares, internal port scanning, remote code execution, and denial of service attacks.*
5. ***Broken Access Control****. Restrictions on what authenticated users are allowed to do are often not properly enforced. Attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users' accounts, view sensitive files, modify other users' data, change access rights, etc.*
6. ***Security Misconfiguration****. Security misconfiguration is the most commonly seen issue. This is commonly a result of insecure default configurations, incomplete or ad hoc configurations, open cloud storage, misconfigured HTTP headers, and verbose error messages containing sensitive information. Not only must all operating systems, frameworks, libraries, and applications be securely configured, but they must be patched/upgraded in a timely fashion.*

7. ***Cross-Site Scripting XSS***. *XSS flaws occur whenever an application includes untrusted data in a new web page without proper validation or escaping, or updates an existing web page with user-supplied data using a browser API that can create HTML or JavaScript. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.*
8. ***Insecure Deserialization***. *Insecure deserialization often leads to remote code execution. Even if deserialization flaws do not result in remote code execution, they can be used to perform attacks, including replay attacks, injection attacks, and privilege escalation attacks.*
9. ***Using Components with Known Vulnerabilities***. *Components, such as libraries, frameworks, and other software modules, run with the same privileges as the application. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications and APIs using components with known vulnerabilities may undermine application defenses and enable various attacks and impacts.*
10. ***Insufficient Logging & Monitoring***. *Insufficient logging and monitoring, coupled with missing or ineffective integration with incident response, allows attackers to further attack systems, maintain persistence, pivot to more systems, and tamper, extract, or destroy data. Most breach studies show time to detect a breach is over 200 days, typically detected by external parties rather than internal processes or monitoring.*

(5 marks)

b) [Syllabus coverage 6.2]

*A detailed explanation of one of the top 10 vulnerabilities (i.e **Cross-Site Scripting XSS**. XSS flaws occur whenever an application includes untrusted data in a new web page without proper validation or escaping, or updates an existing web page with user-supplied data using a browser API that can create HTML or JavaScript. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.)*

(5 marks)

c) [Syllabus coverage 6.1]

***Basic**: a short description of threat modelling (i.e. "threat modeling is the use of abstractions to aid in thinking about risks." Or Threat modelling works to identify, communicate, and understand threats and mitigations within the context of protecting something of value.*
***Stronger**: above plus mentioning why we do threat modelling (i.e. mentioning some of the following points:*

- *Build a secure design*
- *Efficient investment of resources; appropriately prioritize security, development, and other tasks*
- *Bring Security and Development together to collaborate on a shared understanding, informing development of the system*

- *Identify threats and compliance requirements, and evaluate their risk*
- *Define and build required controls.*
- *Balance risks, controls, and usability*
- *Identify where building a control is unnecessary, based on acceptable risk*
- *Document threats and mitigation*
- *Ensure business requirements (or goals) are adequately protected in the face of a malicious actor, accidents, or other causes of impact*
- *Identification of security test cases / security test scenarios to test the security requirements*

***Exemplary**: Mentioning the threat modelling process and the main 4 questions that most threat model methodologies try to answer:*

1. What are we building?

As a starting point you need to define the scope of the Threat Model. To do that you need to understand the application you are building, examples of helpful techniques are:

- *Architecture diagrams*
- *Dataflow transitions*
- *Data classifications*
- *You will also need to gather people from different roles with sufficient technical and risk awareness to agree on the framework to be used during the Threat Modelling exercise.*

2. What can go wrong?

This is a "research" activity in which you want to find the main threats that apply to your application. There are many ways to approach the question, including brainstorming or using a structure to help think it through. Structures that can help include STRIDE, Kill Chains, CAPEC and others.

3. What are we going to do about that?

In this phase you turn your findings into specific actions.

4. Did we do a good enough job?

Finally, carry out a retrospective activity over the work you have done to check quality, feasibility, progress, and/or planning.

(15 marks)

END OF EXAMINATION