MANIPAL UNIVERSITY JAIPUR
University under Section 2(f) of the UGC Act

Centre for Distance & Online Education

INTERNAL ASSIGNMENT

SESSION	**APRIL2025**
PROGRAM	**BACHELOR OF COMPUTER APPLICATIONS (BCA)**
SEMESTER	**VI**
COURSE CODE & NAME	**DCA3241**
CREDITS	**4**
NUMBER OF ASSIGNMENTS & MARKS	**02** **30 marks for each**

Please read the below instructions carefully before proceeding further:

- Learners are instructed to download the IA Question Paper, prepare the answers (Soft Copy), and submit them through Learning Management System (LMS) Portal
- **The last IA assignment submission date (Set-1 & Set-2 in a single file) is reflected in LMS only. This is the last date, and no further extension will be considered.**
- **Assignment submissions are accepted only in .pdf format.**
- Assignments must be **typed** and **formatted** as per the following specifications:
  - Page Margin – 1 inch on all sides
  - Page Orientation – Portrait
  - Page Size – A4
  - Font Family - Times New Roman
  - Font size - 12
  - Alignment - Justified.
- The total page limit shall not exceed 12 pages.
- **Answers for 10-mark questions should be approximately 400-500 words and not more than 200-250 words for 5-mark questions.**
- The average of both assignments' marks scored by the learner will be considered Internal Assessment Marks.
- Only ONE submission is allowed per assignment.
- Please restrict the assignment document size to <2 MB. Avoid inserting images of very high resolution into the document to remain within the size limit. The assignment response document should NOT contain color images or highlighting of text content.
- Upon successfully submitting IA in LMS, learners can verify the document submitted against each course using the preview tab. If the file submitted has been corrupted or the wrong document submitted, it will not be considered for evaluation.
- If the learner resubmits the assignment, it is permissible only on or before the cut-off date, and the last submission will be considered for evaluation purposes.
- **Content that has been directly copied from the Internet/SLM and Assignments that have been copied and shared among students will be automatically rejected and disqualified.**

Mail id: *helpdesk@mujonline.edu.*
Phone: *+91 79966 00444 (Toll Free)*

SET-I			
Q. No	**Questions**	**Marks**	**Total Marks**
1.	Describe the XML tree structure and explain how elements are nested within each other. What are the common tools and validators used in XML programming? Mention at least three	10	**10**
2.a	What is SOAP, and how is it related to XML and database	**5**	**10**
2.b	Write a simple AJAX code snippet to fetch data from a server and update a webpage without reloading it	**5**	
3.	Explain how the MobiHealth application can use local storage for better user experience.	**10**	**10**
SET-II			
Q. No	**Questions**	**Marks**	**Total Marks**
4.	Discuss the new input types introduced in HTML5 and their significance	**10**	**10**
5.	What is the difference between SSL and TLS, and why has TLS largely replaced SSL	**10**	**10**
6.	What are the limitations of using HTML5 for mobile application development? How can these be mitigated?	**10**	**10**

********** SET – I **********

Q1. Describe the XML tree structure and explain how elements are nested within each other. What are the common tools and validators used in XML programming? Mention at least
Three

Ans:- **XML Tree Structure:**
XML (eXtensible Markup Language) represents data in a hierarchical tree structure where:

The document starts with a single root element.

Every XML document has a parent-child relationship between elements.

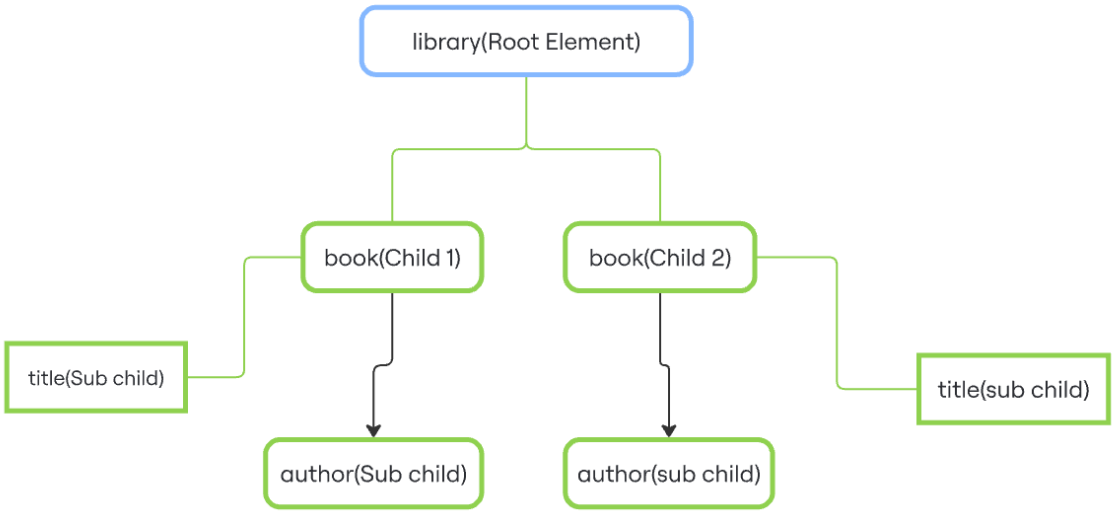
Each element can contain sub-elements (children), attributes, and text content.

Example:

Mail id: *helpdesk@mujonline.edu.*
Phone: *+91 79966 00444 (Toll Free)*

```
<library>
  <book>
    <title>XML Basics</title>
    <author>John Doe</author>
  </book>
  <book>
    <title>Advanced XML</title>
    <author>Jane Smith</author>
  </book>
</library>
```

Tree Representation:



Common Tools & Validators in XML Programming

1) **XML Validator (online tools like XMLValidation.com):**

- Validates well-formed XML.
- Can check against DTD or XSD schemas.

Mail id: *helpdesk@mujonline.edu.*
Phone: *+91 79966 00444 (Toll Free)*

2) . XMLSpy (Altova):

- A professional-grade XML editor.
- Offers schema design, validation, XPath/XQuery support, and visual tree view.

3) xmllint (Command-line tool):

- Comes with libxml2.
- Validates XML syntax, formats XML, and checks against schemas.

Q2

a) What is SOAP, and how is it related to XML and database.

Ans –

What is SOAP?

SOAP (Simple Object Access Protocol) is a protocol used for exchanging structured information in the implementation of web services over a network. It is platform-independent and language-neutral, allowing different systems to communicate using standardized XML messages.

Relationship between SOAP, XML, and Databases

1. SOAP and XML:

a) SOAP uses XML as its message format.
b) Every SOAP message is an XML document containing:
c) An envelope (defines the start and end of the message).
d) An optional header (for metadata or authentication).
e) A body (contains the actual data or function call).

Example SOAP Message:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Body>
    <getBookDetails>
      <bookId>101</bookId>
    </getBookDetails>
  </soap:Body>
</soap:Envelope>
```

2. SOAP and Database:

a) SOAP is commonly used in web services to allow client applications to interact with a server's database.
b) A SOAP-based web service acts as a middle layer:
c) It receives XML requests.
d) Processes data (e.g., using SQL queries to access a database).
e) Sends back XML responses with results.

Example Flow:

Mail id: *helpdesk@mujonline.edu.*
Phone: *+91 79966 00444 (Toll Free)*

```
Client App → SOAP Request (XML) → Server → Query Database → Response
(XML) → Client
```

Q2.

b) <u>Write a simple AJAX code snippet to fetch data from a server and update a webpage without reloading it</u>

Ans –

simple AJAX code snippet using vanilla JavaScript (no libraries like jQuery) to fetch data from a server and update the webpage without reloading:

HTML

```
<!DOCTYPE html>
<html>
<head>
  <title>AJAX Example</title>
</head>
<body>
  <h2>User Info</h2>
  <button onclick="loadUser()">Load User</button>
  <div id="userInfo">User details will appear here.</div>

  <script src="script.js"></script>
</body>
</html>
```

<u>JavaScript (script.js)</u>

```
function loadUser() {
  const xhr = new XMLHttpRequest();
  xhr.open("GET", "https://jsonplaceholder.typicode.com/users/1",
true);

  xhr.onload = function () {
    if (xhr.status === 200) {
      const user = JSON.parse(xhr.responseText);
```


Mail id: *helpdesk@mujonline.edu.*
Phone: *+91 79966 00444 (Toll Free)*

```
      document.getElementById("userInfo").innerHTML =
        `<strong>Name:</strong> ${user.name} <br>
         <strong>Email:</strong> ${user.email} <br>
         <strong>City:</strong> ${user.address.city}`;
    } else {
      document.getElementById("userInfo").innerText = "Failed to load
user data.";
    }
  };

  xhr.onerror = function () {
    document.getElementById("userInfo").innerText = "Request error.";
  };

  xhr.send();
}
```

When the user clicks the Load User button:

- It makes a GET request to a mock API.
- Parses the JSON response.
- Updates the content of without reloading the page.

Q3. Explain how the MobiHealth application can use local storage for better user experience.

Ans –

Local Storage is a feature of web and mobile applications that allows data to be stored on the user's device in key-value pairs. It's especially helpful for improving performance, enabling offline access, and reducing server load.

What is Local Storage?

Local storage in browsers (via localStorage) allows you to store up to 5–10 MB of data persistently (even after closing the app or browser), unlike session storage.

Use Cases for MobiHealth Using Local Storage

1. Offline Access to Medical Records

Store recently viewed prescriptions, reports, or appointments in local storage.
When the user is offline, they can still view their last accessed health data.

```
localStorage.setItem("lastReport", JSON.stringify(reportData));
```

2. Remember User Preferences

Save settings like language, theme (dark/light mode), or preferred doctor.
The app will load faster and feel more personalized on every visit.

```
localStorage.setItem("preferredDoctor", "Dr. A. Kumar");
```

3. Quick Login Experience

Store a login token (carefully, and only if security allows).
Allows users to stay signed in, improving usability without repeated logins.

4. Draft Appointment Requests or Forms

If a user starts filling out a form (like symptom checker or appointment booking) and accidentally closes the app, local storage can save the draft automatically.

```
window.addEventListener("beforeunload", function () {
  localStorage.setItem("draftAppointment",
JSON.stringify(appointmentData));
});
```

********** SET – II **********

Q. Discuss the new input types introduced in HTML5 and their significance.

Ans –

New Input Types in HTML5 and Their Significance

HTML5 introduced several new input types to enhance form handling, improve user experience, and reduce the need for custom JavaScript validation. These new types help ensure that data is entered correctly and efficiently, especially on mobile devices.

Significance of HTML5 Input Types

1. Improved User Experience

- Mobile browsers show appropriate keyboards (e.g., number pad for tel, date picker for date).
- Reduces user errors and speeds up form filling.

2. Built-in Validation

-- Browsers automatically validate formats (e.g., invalid emails won't pass).

-- Reduces reliance on JavaScript for basic checks.

3. Better Accessibility
-- Screen readers and accessibility tools recognize these types and provide better context to users with disabilities.

4. Cleaner Code & Less JavaScript
-- Developers can avoid writing repetitive scripts for format validation or UI elements like date pickers.

5. Cross-Platform Consistency
-- Uniform behavior across browsers and devices helps create consistent UI/UX patterns.

Example:

```
<form>
  <label>Email: <input type="email" required></label><br>
  <label>Birth Date: <input type="date"></label><br>
  <label>Favorite Color: <input type="color"></label><br>
  <button type="submit">Submit</button>
</form>
```

Q5. What is the difference between SSL and TLS, and why has TLS largely replaced SSL

Ans

What is a Secure Sockets Layer (SSL)?
A Secure Sockets Layer (SSL) is a cryptographic protocol developed by Netscape in the mid-1990s to secure internet communications. It was designed to provide privacy, authentication, and data integrity between web browsers and servers. SSL quickly became the standard for securing online transactions, such as e-commerce and online banking, due to its ability to encrypt data and ensure secure connections.

Despite its initial success, SSL had inherent weaknesses that made it susceptible to various attacks. In September 2014, Google discovered a serious SSL 3.0 vulnerability called Padding Oracle on Downgraded Legacy Encryption (POODLE), which hackers exploited to decrypt and

Mail id: *helpdesk@mujonline.edu.*
Phone: *+91 79966 00444 (Toll Free)*

steal confidential information. These vulnerabilities led to the development of TLS as a more secure successor to SSL.

What is Transport Layer Security (TLS)?

Transport Layer Security (TLS) is the successor to SSL, developed primarily to address the security vulnerabilities found. Originally introduced in 1999, and now on version 1.3, TLS provides enhanced security features and improved performance, making it the modern standard for secure internet communications. At its core TLS serves the same purpose as SSL - ensuring the privacy, integrity, and authenticity of data exchanged between web browsers and servers.

Key Differences Between SSL and TLS

The key difference between SSL and TLS is the introduction of several key improvements, including stronger encryption algorithms, better authentication mechanisms, and more robust key exchange methods. These enhancements significantly reduce the risk of attacks and ensure a higher level of security for data transmission. Some of the notable features and differences of TLS include:

Stronger Encryption: TLS supports advanced encryption algorithms, such as AES (Advanced Encryption Standard), which provides better protection against brute-force attacks.

Cryptographic Differences: While both protocols use similar concepts, TLS incorporates more robust cryptographic techniques. For example, TLS uses HMAC (Hash-based Message Authentication Code) for message integrity, which is more secure than the MAC (Message Authentication Code) used in SSL. Additionally, TLS supports more secure cipher suites and provides better protection against cryptographic attacks. Learn more about Hash-based cryptography with our support article.

Improved Authentication: TLS uses more secure methods for verifying the identity of communicating parties, reducing the risk of man-in-the-middle attacks. The TLS handshake process is more secure and allows for robust authentication mechanisms.

Performance Improvements: TLS also brings performance improvements over SSL. The protocol is designed to be more efficient, reducing the overhead associated with establishing secure connections. This results in faster handshake processes and lower latency, making TLS more suitable for modern web applications that require high performance and low response times.

Enhanced Key Exchange: TLS employs more secure key exchange protocols, such as Diffie-Hellman and Elliptic Curve Diffie-Hellman, to establish secure connections.

6. What are the limitations of using HTML5 for mobile application development? How can these be mitigated?
Ans –

Limitations of Using HTML5 for Mobile Application Development — and How to Mitigate Them

While HTML5 has been widely adopted for building cross-platform mobile applications, it comes with several limitations that can affect performance, user experience, and native integration. Below is a detailed and professional explanation of these limitations, along with practical ways to mitigate them.

Key Limitations of HTML5 in Mobile App Development

1. Limited Access to Native Device Features
HTML5 apps running in a browser or WebView may lack deep access to native device features such as GPS, camera, Bluetooth, accelerometer, fingerprint scanner, etc.

Mitigation:
Use hybrid frameworks like Apache Cordova, Capacitor, or Ionic that provide JavaScript APIs to bridge HTML5 with native device capabilities.

2. Performance Constraints
HTML5 apps may perform slower than native apps, especially for graphics-intensive tasks like animations, real-time games, or video processing.

Mitigation:
Optimize code with Web Assembly, efficient rendering, and consider using progressive enhancement. For highly demanding apps, use native code or cross-platform solutions like Flutter or React Native.

3. Inconsistent User Experience Across Devices
Different mobile browsers have varying levels of support for HTML5 features, which can lead to inconsistent rendering or behavior.

Mitigation:
Perform extensive cross-browser testing using tools like Browser Stack or Sauce Labs, and use feature detection with libraries like Modernizer to implement fallbacks.

4. Offline Functionality Limitations

Mail id: *helpdesk@mujonline.edu.*
Phone: *+91 79966 00444 (Toll Free)*

HTML5 supports local storage and caching (via local Storage, Indexed DB, and service workers), but the offline capabilities are still limited compared to native apps.

Mitigation:
Implement Progressive Web Apps (PWAs) which leverage service workers, cache APIs, and background sync to improve offline support and app-like behavior.

5. Security Concerns
Being web-based, HTML5 apps are more exposed to web security threats, such as XSS, CSRF, and code injection.

Mitigation:
Follow best practices for web security:

- Sanitize user inputs
- Use HTTPS
- Implement CORS policies
- Regularly audit code using tools like OWASP ZAP or Snyk

6. Limited App Store Visibility & Functionality
Pure HTML5 apps may not qualify for app store distribution or lack monetization and notification features.

Mitigation:
Wrap HTML5 apps using Cordova, Capacitor, or Tauri to generate installable app packages for App Store and Google Play distribution.

7. Longer Initial Load Times
HTML5 apps often load over the web, which can lead to slower start-up times compared to pre-installed native apps.

Mitigation:
Use lazy loading, code splitting, and asset compression (like Brotli or GZIP) to reduce initial load time.

Mail id: *helpdesk@mujonline.edu.*
Phone: *+91 79966 00444 (Toll Free)*